1971-12

The modeling and analysis of an elastic mechanism with clearances

Gnilka, Charles William

Monterey, California ; Naval Postgraduate School

THE MODELING AND ANALYSIS OF AN ELASTIC MECHANISM WITH CLEARANCES

CHARLES WILLIAM GNILKA

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

THE MODELING AND ANALYSIS OF AN ELASTIC
MECHANISM WITH CLEARANCES

by

Charles William Gnilka

Thesis Advisor: D. Salinas

December 1971

Approved for public release; distribution unlimited.

The Modeling and Analysis of An Elastic
Mechanism With Clearances

by

Charles William Gnilka
Lieutenant, United States Navy
B.S., Villanova University, 1964

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1971

ABSTRACT

The classical approach for the analysis of linkages has been to assume rigid elements during operation. As the operational speeds of mechanisms increase, this assumption creates larger errors in the analysis. Thus a better model of a machine that considers elastic deformations will aid in more efficient design as well as provide improved accuracy and performance.

This thesis presents a simulation for the dynamic response of an elastic link model with three clearances. The simulation has been developed for an automobile, cam actuated valve train. The computer program is coded in FORTRAN IV, for an IBM-360 computer and is included as a portion of this work. The capability to visually observe the dynamic action of the model is included by a graphic display routine. This routine is implemented in SDS FORTRAN IV for a SDS-9300 computer interfaced with an ADAGE Graphic Display Terminal, Model 10.

Sample problems for various cam speeds of interest are included utilizing graphs and photos from the graphic display.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF PHOTOGRAPHS

LIST OF SYMBOLS

A	Area
$[C]$	System damping matrix
c_{ij}	Element of damping matrix
E	Elastic modulus
F_i	Forces
$\{F^*\}$	Normalized force vector
g	Gravitational Constant
I	Moment of inertia
$[K]$	System stiffness matrix
k_{ij}	Element of stiffness matrix
L	Length
$[M]$	System mass matrix
M	Mass
m_{ij}	Elements of mass matrix
$\{p\}$	Force vector used in integration solution
q	System coordinate
t	Time
ΔT	Time increment used in integration solution
T_i	Period
u	Element coordinate
$\{\alpha\}$	Vector used in integration solution
$[\beta]$	Transformation matrix
ζ	Damping coefficient
	Cam angle of rotation

$\{\Phi\}$	Eigenvectors listed in columns, modal matrix
$\{\Psi\}$	Vector used in integration solution
ω_n, Ω	Natural frequency
$< >$	Row Vector

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to his advisors, Dr. Richard C. Winfrey currently at the Naval Civil Engineering Laboratory, Port Hueneme, who introduced this author to the project and for his constructive supervision; and Professor David Salinas of the Naval Postgraduate School for his moral support and guidance.

The Naval Postgraduate School Computer Center provided facilities for the computer work, except the computer graphic portion. This portion was accomplished with facilities and technical advice provided by the Electrical Engineering Department Computer Laboratory. In this regard, the author wishes to express his sincere thanks to Mr. Robert L. Limes, computer laboratory Supervisor, and the staff Mr. William A. Thomas and Mr. Albert Wong for the hours in assistance with graphic computer programing and hardware difficulties.

Finally, the author wishes to acknowledge the encouragement, patience, and moral support of his wife, Martha and for her assistance in preparing the manuscript.

I. INTRODUCTION

The classical approach to the dynamic analysis of linkages has been to assume that all menbers of the machine move without deformation during operation, i.e. that they move as rigid bodies. Based on this assumption, material choices and sizing to withstand stresses for the applied loads, the linkage deformations and the response motion have been severly idealized. Application of factors of safety have been necessary to account for this idealization. At lower operational speeds design criteria are met. However as the speeds increase the resulting large inertial forces and elastic deflections are more severe and thus the possibility of failure more acute. The design becomes invalid when the critical speed is reached where the machine either fails or where it no longer performs its intended task. Thus expensive and time consuming models must be built and tested, for elastic deformations over the range of operational speeds before the kinematic analysis can be fully accepted. The goal then is to obtain a more accurate dynamic model of a machine in order to better predict and understand its performance.

With the advent of the digital computer, more complex mathematical models of machine are being developed. Engineers are able to accomplish more refined designs with a higher degree of operational confidence, less of a factor of safety, better material choice, and fewer physical model experiments.

This work presents a simulation for the dynamic response of a planar elastic link mechanism, free to move in its rigid body motion

[illegible]

as well as its elastic motion. The structural elements deform elastically due to forces applied by neighboring elements, and due to inertial forces. This simulation has been developed for an automobile, cam actuated valve train, with three clearances. The clearances are considered to exist at the (1) cam-pushrod, (2) valve-valve seat, and (3) rocker arm-pushrod. To simplify the required derivations, the model is assumed to initially consist of straight members with constant circular cross-sectional bars. The assumption of constant circular cross-sectional bars is selected for ease in computing beam section properties. The use of straight links greatly simplifies the equations of elastic deformation.

An existing computer program written by Anderson and Winfrey (Ref. 2), was dependent on separately coded sections for each physical configurations of the two clearance model of a cam actuated valve train. As the number of system clearances are increased, the number of system configurations increase exponentially and separately coded program sections become infeasible.

The extent of this author's objectives and work on the existing program is to:

(1) Restructure and reprogram the simulation to provide for a generalized method for establishing the physical model configurations.

(2) Modify this program to allow for more than two clearances, i.e. add clearances internal to the system.

(3) Establish a method for the dynamic graphical display of the model showing the resulting elastic deformations of the model members at high operational speeds.

Objectives (1) and (2) have been accomplished using the IBM-360 computer for the simulation. Objective (3) has been accomplished with the aide of the ADAGE Graphics Terminal model 10 (AGT-10) interfaced with a Scientific Data System computer model 9300 (Photo 1.1). Data generated by the IBM computer is loaded into the SDS computer by means of punched cards or magnetic tape. This data is then manipulated into a form acceptable for display by the AGT-10.

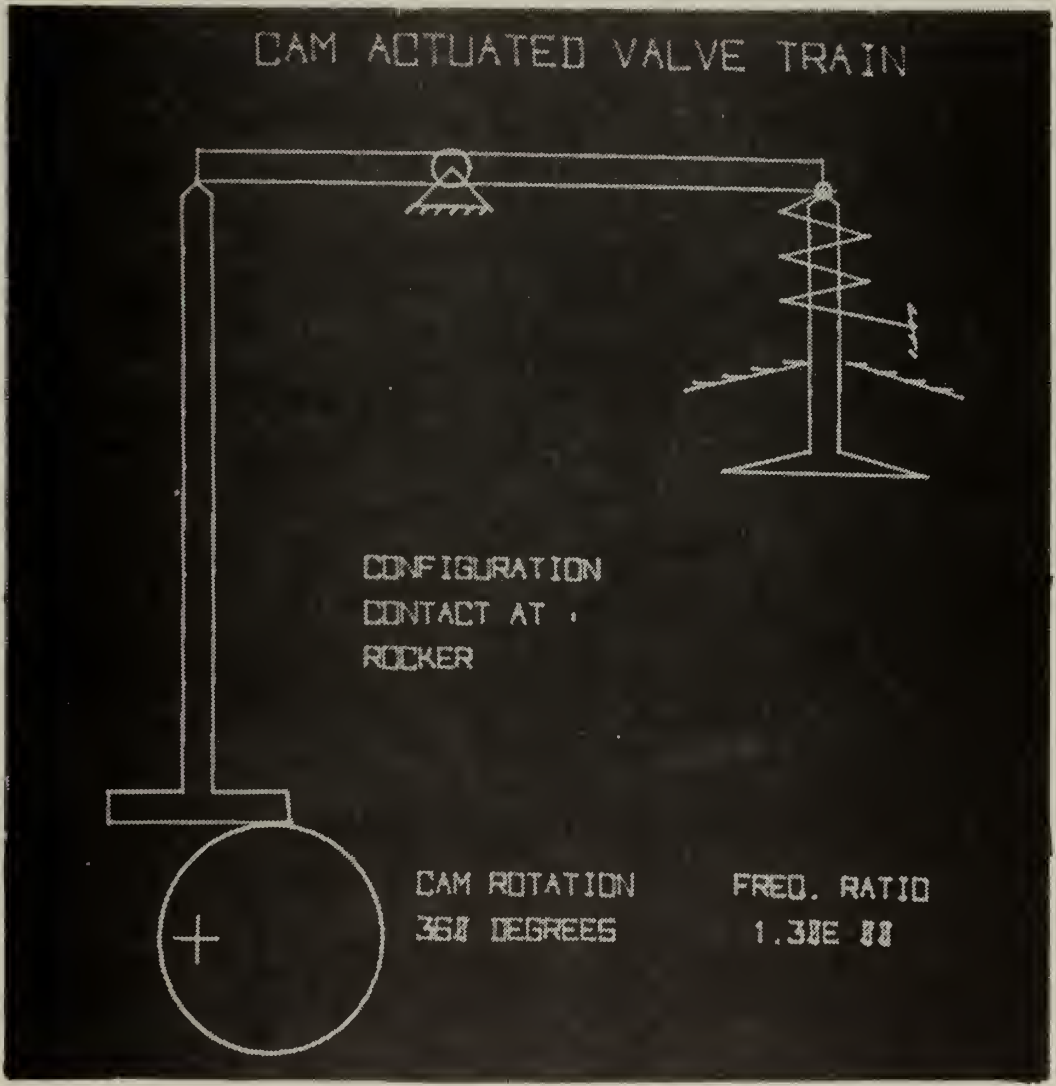


Photo 1.1 AGT DISPLAY

In the following chapters, the physical model, and the concepts and formulation for the corresponding mathematical model are described. The dynamic analysis and the elastic deformation computer programs are discussed and listed at the end of this work. Finally, several numerical examples are given with the model responses shown as AGT photographs and CALCOMP graphs.

In the following chapters, the physical model, and the concepts and formulation for the corresponding mathematical model are described. The dynamic analysis and the elastic deformation computer programs are discussed and listed at the end of this work. Finally, several numerical examples are given with the model responses shown as AGT photographs and CALCOMP graphs.

II. PHYSICAL MODEL

In this chapter the physical system, Figure 2.1, is idealized to enable formulation of a mathematical model whose analysis is tractable. The description covers the interaction between the system parameters and the physical contraints that limit the model's motion. The physical elements that comprise the system are listed and their functions explained. Additionally their physical properties, size and material content, are noted.

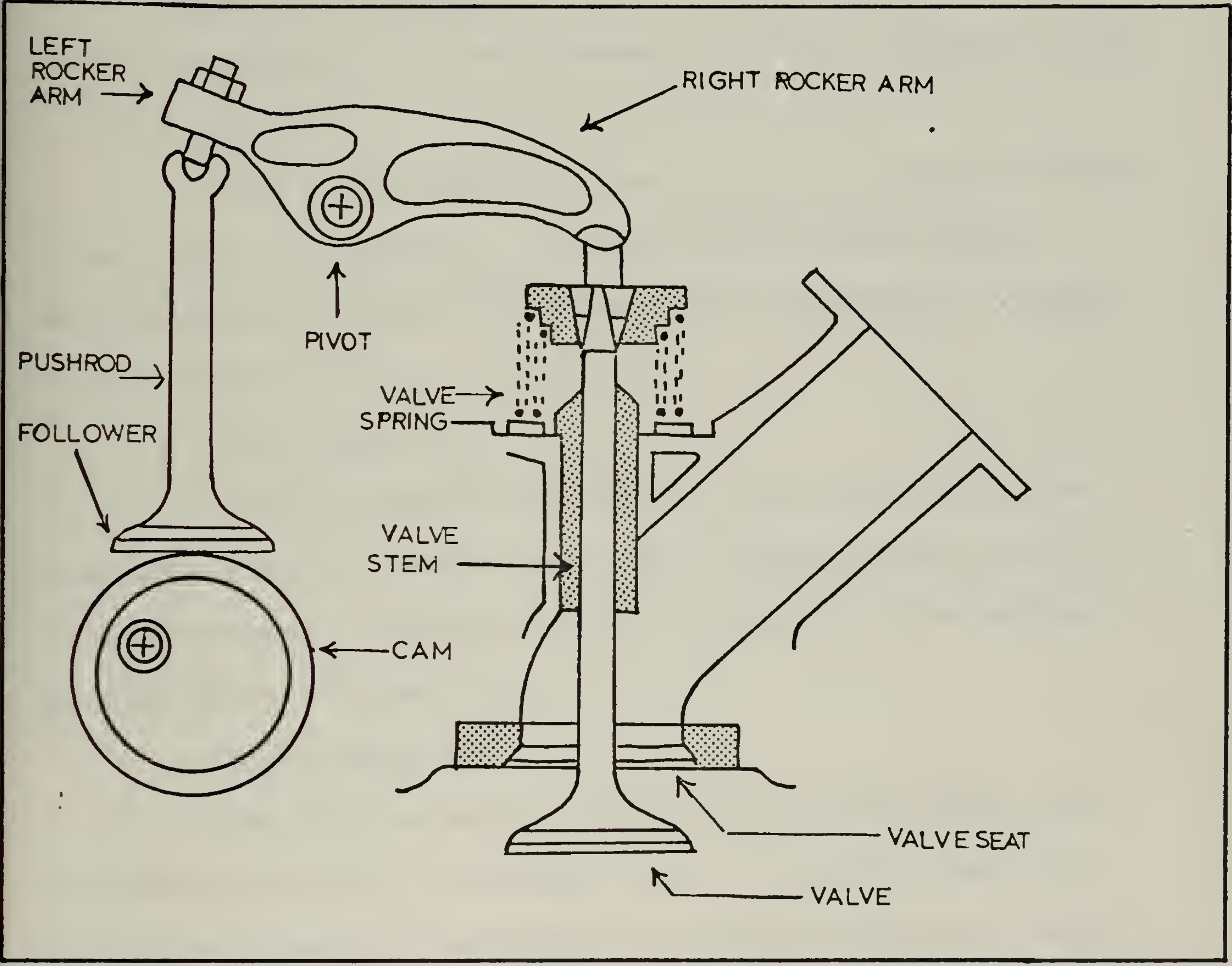


Figure 2.1 CAM ACTUATED VALVE TRAIN

A. GENERAL DESCRIPTION

In order to relate the formulation of this mathematical model to the real world, an automobile car engine valve train assembly is considered. However, any similar physical model might have been chosen. All members of the model are assumed to be manufactured from steel with an elastic modulus of 30×10^6 pounds per square inch and a specific weight of .283 pounds per cubic inch. Additionally longitudinal members are considered as rods with circular cross sections. The diameter and length of each bar are specified.

1. Elements of the Model

The elements of the valve assembly pictured in Figure 2.1 are:

a. Follower

This reciprocating flat faced follower is rigidly attached to the pushrod. The pushrod is considered as an elastic member. These members have harmonic displacement. The diameter is .3 inches; length 9 inches.

b. Left Rocker Arm

The left rocker arm is pinned to and rotates about the fixed pivot point at its right end. This element provides for an "internal" clearance between itself and the pushrod at its left end. The diameter is .6 inches; length 1 inch.

c. Right Rocker Arm

The right rocker arm is also pinned to and rotates about the fixed pivot point. At its right end it is pinned to the valve stem and valve spring. The diameter is .6 inches; length 1.5 inches.

d. Valve Spring

The free end of the valve spring is connected to the rocker arm and valve stem pin, the other end is rigidly fixed to ground. The spring is assumed to have a spring constant of 245 pounds per inch. The diameter is .07 inches; free length 36 inches.

e. Valve

This reciprocating valve is rigidly attached to the valve stem. The valve stem is assumed to be elastic. The upper end of the valve stem is connected to the common pin with the valve spring and right rocker arm. The diameter is .3 inches; length 3 inches.

To complete the model two additional elements are added external to the system. Both provide for the "external" clearances. They are:

f. Cam

The offset circular disk cam imparts sinosudial motion to the follower.

g. Valve Seat

This rigid seating surface for contact by the valve may be removed from the system when desired.

The rocker arm, although modeled as two elements, is considered to be one continuous member. Thus only rotation about the fixed pivot point is permitted.

2. Clearance Parameters

The clearance or interfereance at the valve seat may be eliminated by the introduction of a predetermined parameter. Due to the choice of the coordinate system, the parameter set less than the minimum value of the valve displacement has the effect of removing

the valve seat from the system. This provides for the flexibility to conduct studies of the valve motion unimpeded by the valve seat.

Additional flexibility of the model is provided by introduction of another clearance parameter. This parameter has the effect of eliminating the internal clearance between the pushrod and left rocker arm. Thus these two members can be pinned together when desired.

The circular disk cam rotates counterwise. This motion first lifts the follower by direct contact to its extreme position and next permits the follower to return to its initial position. The harmonic cycle is then repeated. For systems with cams of this type, the return motion of the follower must be accomplished by some force external to the cam. Therefore the valve spring is introduced to supply this force. For this work, simple harmonic motion is specified for the follower. The follower then oscillates about the midstroke position with an acceleration proportional to its distance from this position. The amplitude is half the stroke. The cam profile used to obtain the harmonic motion has been generated by the equation $.5 \sin(\omega t)$ inches, where ω is the cam speed in radians per second. Similar studies of various other cam profiles are discussed in Refs. 14-18.

B. ELEMENT COORDINATES

The model of the mechanism consists of two basic elements which account for all the relevant energy. The axial element, Figure 2.2 considers tension and compression forces. The bending element, Figure 2.3, considers only bending energy. The coordinates of the five

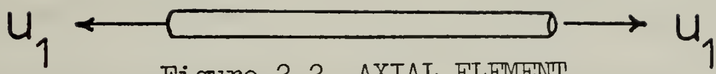


Figure 2.2 AXIAL ELEMENT

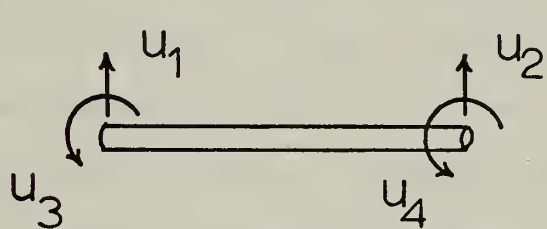


Figure 2.3 BENDING ELEMENT

elements of the system, i.e. pushrod, rocker arms, valve, and spring are chosen as indicated in Figure 2.4.

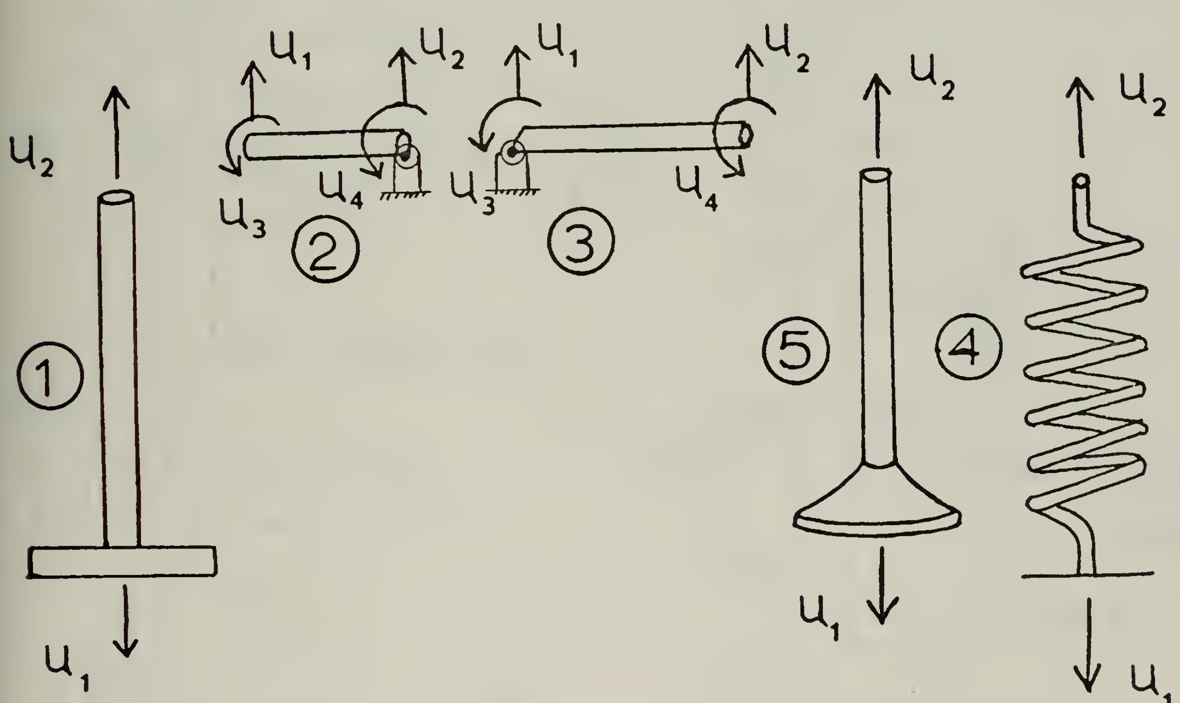


Figure 2.4 COMBINED ELEMENT COORDINATES

C. SYSTEM COORDINATES

Assuming small deformation theory, the linear model is composed of the above elements by superposition. The external members, the cam and valve seat, are also assigned coordinates for reference. As noted above, the pivot point and lower end of the spring are connected to ground and are therefore rigid. The systems coordinates {q} define motion. Since no motion is assumed for these two connections, no coordinates are necessary. Recalling that each member was assumed straight, the system coordinates will generally coincide with the element coordinates. The coordinate indices and their positive senses are shown in Figure 2.5.

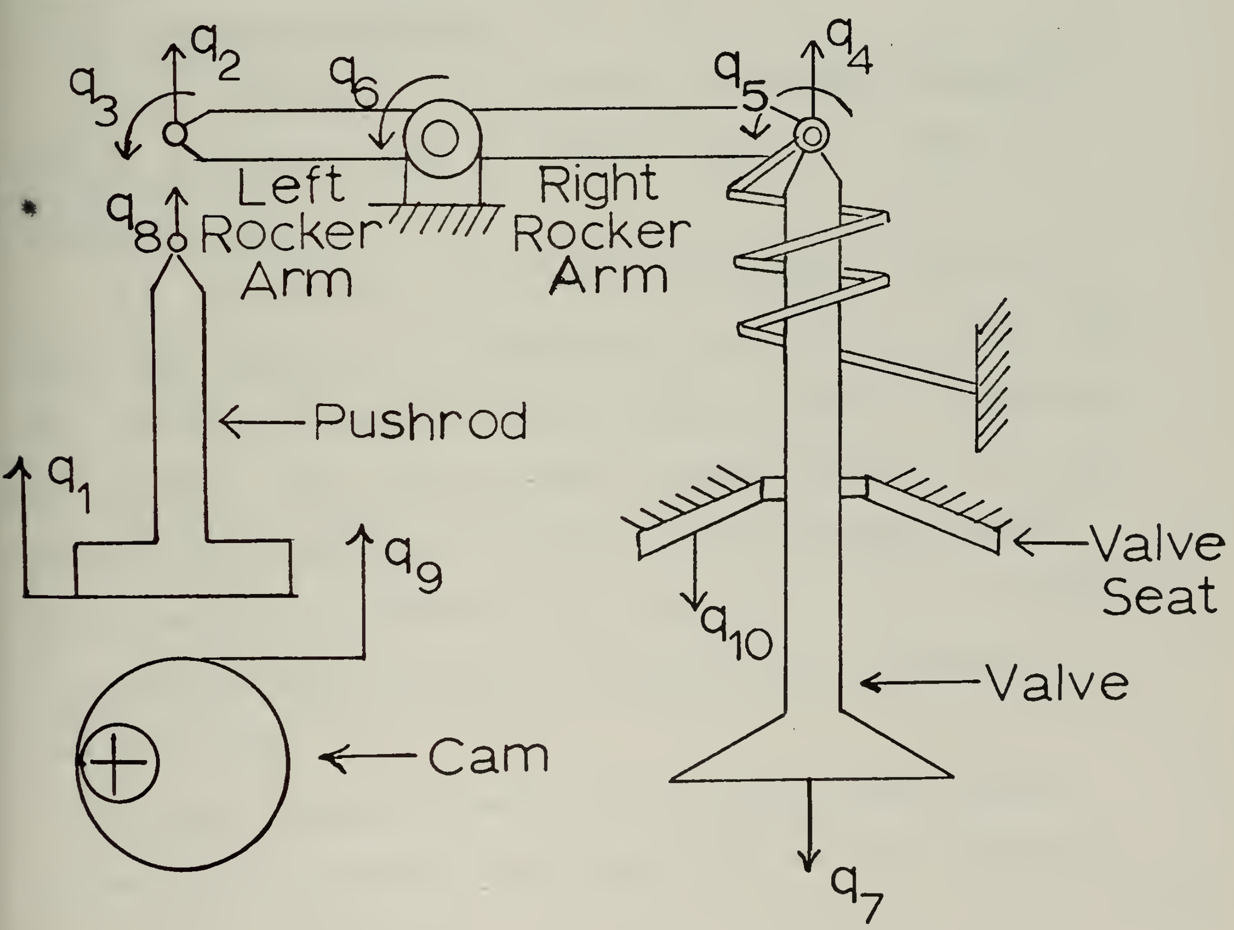


Figure 2.5 FULL SYSTEM COORDINATES

D. DEFINITIONS OF CONFIGURATIONS

The model shown in Figure 2.5 has two external clearances and one internal clearance. The term external refers to a displacement difference between a member of the system and an external member. Internal clearances are displacement differences between two internal members of the system. The clearances are:

1. <u>External To The System</u>

   a. Clearance 1. The displacement difference between the cam, (coordinate nine), and the follower, (coordinate one).

   b. Clearance 3. The displacement difference between the valve, (coordinate seven), and the valve seat, (coordinate ten).

2. <u>Internal To The System</u>

   a. Clearance 2. The displacement difference between the pushrod, (coordinate eight) and left end of the left rocker arm, (coordinate two).

   b. Clearance 4. The displacement difference between the valve stem, free end of the spring, and right end of the right rocker arm. The basic concept for the four clearance problem is presented in Appendix C. Since only the three clearance problem is considered for this work, clearance four is zero. Thus the members remain pinned and all move together at this connection. Only one coordinate is required to model the motion of the joint and is designated coordinate four.

Thus the external clearances define positions where external forces must be considered, i.e. forces associated with coordinate

one and coordinate seven. The internal clearances produce internal forces. These equal and opposite forces act upon the coordinate pair two and eight only when they are in contact.

The three clearances, where positive displacement differences between the coordinate pairs may exist, will be referred to as noted above, i.e. clearances 1, 2, and 3. Since clearance 4 will not be modeled, reference to this position will be referred to as coordinate four. Forces between clearances will be discussed in section 3.E.

E. SYSTEM CONFIGURATION

Depending upon the value of the force and displacement at these three clearances, the model can exist in one of eight possible configurations. If clearance 2 is considered pinned, the difference between coordinates two and eight is zero. The possible configurations is then reduced to four and is the study conducted by Anderson (Ref. 2).

The eight configurations are diagramed in Table 2.1 and 2.2. They are:

1. No contact at clearance 1, clearance 2, or clearance 3.
2. Contact at clearance 3; no contact at clearance 1, or clearance 2.
3. Contact at clearance 2; no contact at clearance 1, or clearance 3.
4. Contact at clearance 2, and clearance 3; no contact at clearance 1.
5. Contact at clearance 1; no contact at clearance 2, and clearance 3.

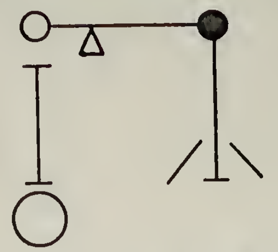
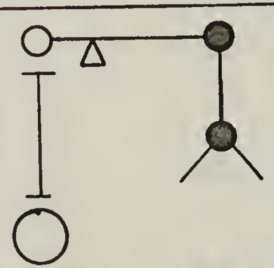
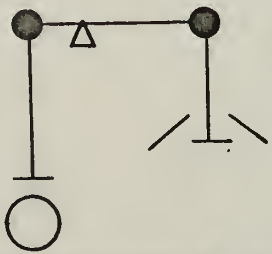
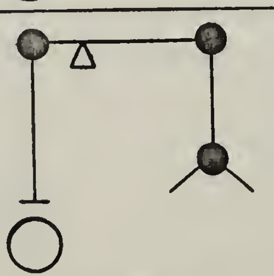
CONFIGURATION		CONTACT AT CLEARANCE	MONTORED VALUES		ASSUMED VALUES	
No.	Figure		Force	DISPL DIFF	Force	Displacements
I		NONE (NO CONTACT)	--	$q_1 - q_9$ $q_2 - q_8$ $q_7 - q_{10}$	$F_1=0$ $F_2=0$ $F_7=0$ $F_8=-F_2$	ALL VALUES CALCULATED
II		3 (VALVE SEAT)	F_7	$q_1 - q_9$ $q_2 - q_8$	$F_1=0$ $F_2=0$ $F_8=-F_2$	$\ddot{q}_7 = \ddot{q}_{10}$; $\dot{q}_7 = \dot{q}_{10}$; $q_7 = q_{10}$
III		2 (ROCKER)	F_2	$q_1 - q_9$	$F_1=0$	$\ddot{q}_2 = \ddot{q}_8$; $\dot{q}_2 = \dot{q}_8$; $q_2 = q_8$. ∴ Seven Coordinate Reduced System
IV		2 and 3 (VALVE SEAT ROCKER)	F_2 F_7	$q_1 - q_9$	$F_1=0$ $F_8=-F_2$	$\ddot{q}_2 = \ddot{q}_8$; $\dot{q}_2 = \dot{q}_8$; $q_2 = q_8$ ∴ Seven Coordinate Reduced System $\ddot{q}_7 = \ddot{q}_{10}$; $\dot{q}_7 = \dot{q}_{10}$; $q_7 = q_{10}$

TABLE 2.1 CONFIGURATION REFERENCE TABLE

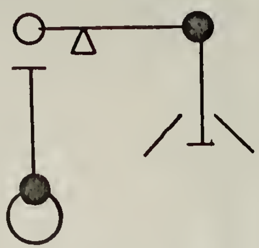
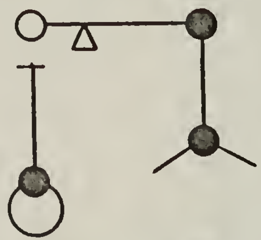
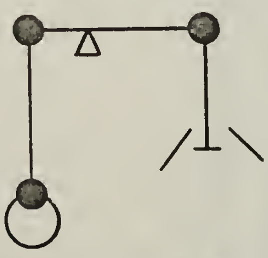
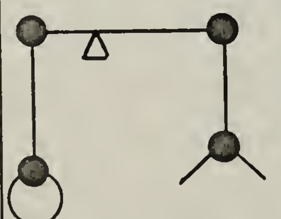
CONFIGURATION		CONTACT AT CLEARANCE	MONTORED VALUES		ASSUMED VALUES	
No.	Figure		Force	DISPL DIFF	Force	Displacements
V		1 (CAM)	F_1	$q_2 - q_8$ $q_7 - q_{10}$	$F_2=0$ $F_7=0$ $F_8=-F_2$	$\ddot{q}_1 = \ddot{q}_9$; $\dot{q}_1 = \dot{q}_9$; $q_1 = q_9$
VI		1 and 3 (CAM VALVE SEAT)	F_1 F_7	$q_2 - q_8$	$F_2=0$ $F_8=-F_2$	$\ddot{q}_1 = \ddot{q}_9$; $\dot{q}_1 = \dot{q}_9$; $q_1 = q_9$ $\ddot{q}_7 = \ddot{q}_{10}$; $\dot{q}_7 = \dot{q}_{10}$; $q_7 = q_{10}$
VII		1 and 2 (CAM ROCKER)	F_1 F_2	$q_7 - q_{10}$	$F_7=0$ $F_8=-F_2$	$\ddot{q}_2 = \ddot{q}_8$; $\dot{q}_2 = \dot{q}_8$; $q_2 = q_8$ $\therefore$ Seven Coordinate Reduced System $\ddot{q}_1 = \ddot{q}_9$; $\dot{q}_2 = \dot{q}_9$; $q_1 = q_9$
VIII		1, 2, and 3 (CAM ROCKER VALVE SEAT)	F_1 F_2 F_7	- - -	$F_8=-F_2$	$\ddot{q}_2 = \ddot{q}_8$; $\dot{q}_2 = \dot{q}_8$; $q_2 = q_8$ $\therefore$ Seven Coordinate Reduced System $\ddot{q}_1 = \ddot{q}_9$; $\dot{q}_1 = \dot{q}_9$; $q_1 = q_9$ $\ddot{q}_7 = \ddot{q}_{10}$; $\dot{q}_7 = \dot{q}_{10}$; $q_7 = q_{10}$

TABLE 2.2 CONFIGURATION REFERENCE TABLE (continued)

6. Contact at clearance 1, and clearance 3; no contact at clearance 2.

7. Contact at clearance 1, and clearance 2; no contact at clearance 3.

8. Contact at clearance 1, clearance 2, and clearance 3.

These eight configurations will be referred to as configurations I through VIII respectively.

The physical constraints of the system require that at any given instant of time, one and only one configuration will be satisfied. The forces and displacement differences at clearances 1, 2, and 3 are monitored to determine which configuration is valid for the calculated values. For example when the model is in a configuration that requires contact at a clearance and the calculated force at this clearance becomes zero (or negative), then contact is lost at that coordinate pair. Similarly, when the model is in a configuration that requires no contact at a clearance and the calculated displacement becomes zero (or negative), an interference exists and contact is regained at the coordinate.

The configuration and the values monitored for the appropriate coordinates are listed in Tables 2.1 and 2.2. Since this model always has contact at coordinate four contact at clearance 2 will be referred to as contact at the "rocker", (between left rocker arm and pushrod).

F. SHIFTING OF CONFIGURATIONS

As discussed above, the model shifts from one configuration to another, depending on first, the configuration the model is presently in; and second, the values of the monitored parameters. This shifting of configurations can be more easily understood by considering the flow diagram in Figure 2.6. The configuration schematic of the two clearance problem of Anderson (Ref. 2) appeared as a square, (face 3-4-8-7). As seen here the schematic of the three clearance problem becomes a cube. The solid lines and adjacent force parameter indicate a shift in the direction of the arrow caused by that particular force parameter. Similarly the dashed lines and adjacent coordinate parameters indicate a shift in the direction of the arrow caused by that particular negative displacement difference. Also indicated in the configuration flow diagram is an impact indicator. When a positive clearance becomes zero (or negative) a rapid change in the associated coordinate acceleration occurs which lends to high forces, i.e. impact. This is further discussed in section 3.F2. Impact is indicated by the rectangular boxes associated with the dashed shift lines. The figures within these boxes are the clearances at which the impact occurs.

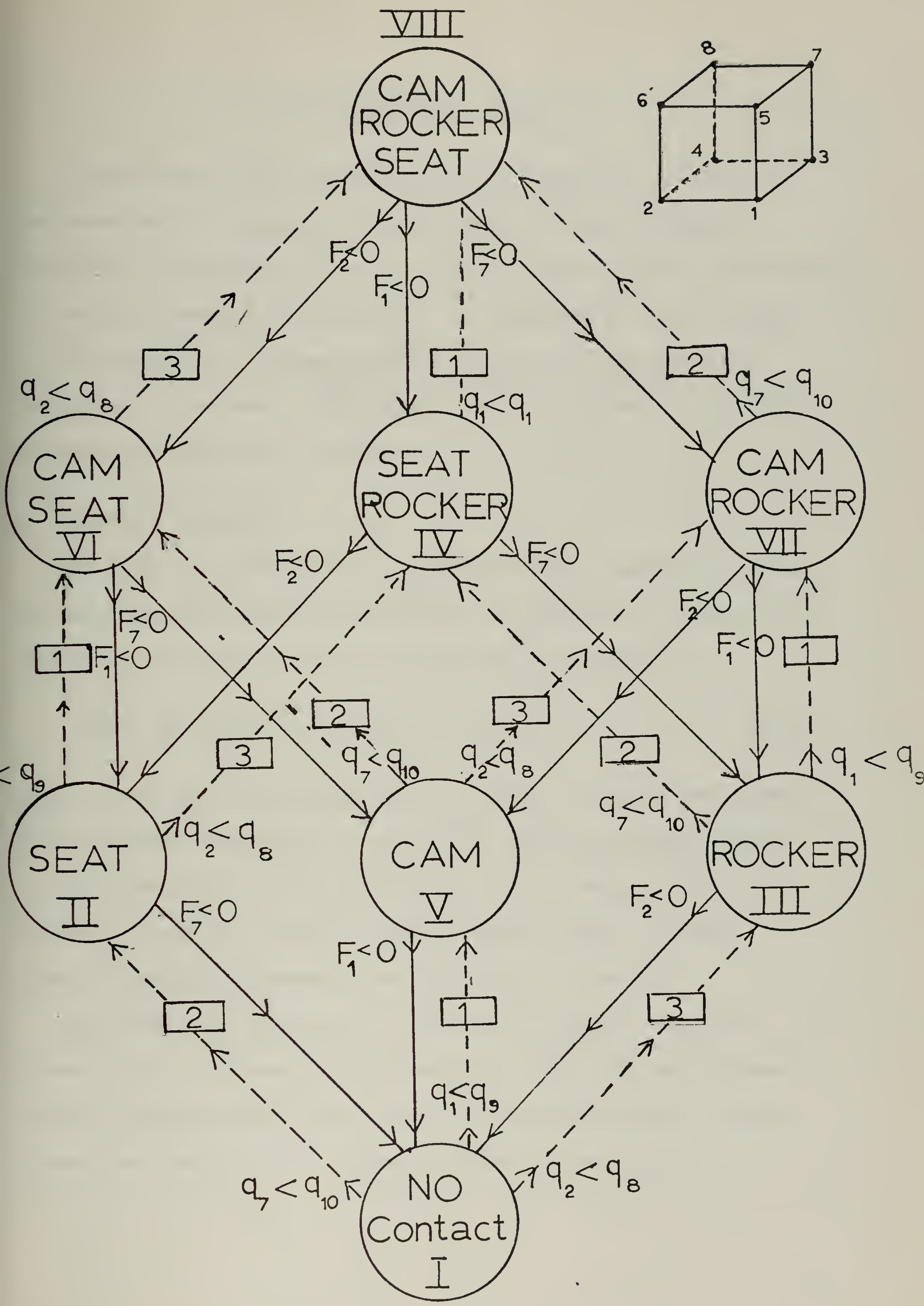


Figure 2.6 FOUR CLEARANCE SYSTEM CONFIGURATION SHIFTING DIAGRAM

III. MATHEMATICAL MODEL

In this chapter the equations of motion which apply to the model are presented and analyzed. The development of the system mass, and stiffness matrices; the two eigenvalue problems; and the formulation of the system damping are discussed. An algorithm to shift the model to the configuration corresponding to the values of the monitored parameters is presented. Finally the numerical methods by which the solutions are obtained is explained.

A. EQUATIONS OF MOTION

The model is considered to be a linear viscous damping spring-mass system, excited by time varying forces. The differential equations of motion can be written in terms of the system coordinates as

$$[M] \; \{\ddot{q}\} + [C] \, \{\dot{q}\} + [K] \, \{q\} \; = \; \{F\} \tag{3.1}$$

where in the usual terminology, [M] represents the system mass matrix, [C] the system damping matrix, and [K] the system stiffness matrix.

The formulation of these matrices has been developed using Lagrange's equations, with the matrix elements based on finite element techniques, Rubenstein and Hurty (Refs. 8 and 9).

The elements of the model, as developed in section 2.B, have a discrete set of element coordinates (u_i = 1,2,....n) that represent the "n", rotational and translational element displacements. These element coordinates {u} are directly related to the system

coordinates $\{q\}$ by

$$\{u\} = [\beta]\{q\} \tag{3.2}$$

The transformation matrix $[\beta]$ is formed by applying the conditions of compatibility to the mechanism. For example, if element coordinate u_4 of element three is rotated by one unit, recalling that linear theory is assumed, it is seen that the relation is

$$(u_4)_3 = q_5 \ . \tag{3.3}$$

This scalar equation viewed as the matrix equation, (3.2), becomes

$$(u_4)_3 = \langle 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0 \rangle \begin{Bmatrix} q_1 \\ q_2 \\ \cdot \\ \cdot \\ q_5 \\ \cdot \\ \cdot \\ q_8 \end{Bmatrix}$$

Thus the fifth row of the transformation matrix $[\beta]$ has been obtained. By similar procedures, the entire transformation matrix for the eight coordinate system is developed. Therefore equation (3.2) can now be written as equation (3.4). Where the column matrix of the element coordinates, $\{u\}$, are partitioned by their appropriate elements, and the 14 x 8 transformation matrix, $[\beta]$ is partitioned into sub-matrices corresponding to the respective elements. A more concise form may be written for equation (3.4) as seen in equation (3.5).

$$
\left\{\begin{array}{l}
\left\{\begin{matrix} u_1 \\ u_2 \end{matrix}\right\}_1 \\
\left\{\begin{matrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{matrix}\right\}_2 \\
\left\{\begin{matrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{matrix}\right\}_3 \\
\left\{\begin{matrix} u_1 \\ u_2 \end{matrix}\right\}_4 \\
\left\{\begin{matrix} u_1 \\ u_2 \end{matrix}\right\}_5
\end{array}\right\}
=
\left[\begin{array}{cccccccc}
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hdashline
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\hdashline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\hdashline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\hdashline
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}\right]
\left\{\begin{matrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \end{matrix}\right\}
\tag{3.4}
$$

$$
\left\{\begin{matrix} \{u\}_1 \\ \{u\}_2 \\ \{u\}_3 \\ \{u\}_4 \\ \{u\}_5 \end{matrix}\right\}
=
\begin{bmatrix} [\beta]_1 \\ [\beta]_2 \\ [\beta]_3 \\ [\beta]_4 \\ [\beta]_5 \end{bmatrix}
\{q\}
\tag{3.5}
$$

B. MASS AND STIFFNESS MATRICES

The element mass $[M_e]$ and stiffness $[K_e]$ matrices are inherent characteristics of each element and can readily be formulated with finite element methods, (Refs. 1 and 8). The system mass $[M]$ and stiffness $[K]$ matrices can be formed with the aid of the transform matrix $[\beta]$ as:

$$[M] = \sum_{i=1}^{5} [\beta]_i^T [M_e]_i [\beta]_i \tag{3.6}$$

$$[K] = \sum_{i=1}^{5} [\beta]_i^T [K_e]_i [\beta]_i \tag{3.7}$$

where $[\beta]_i$ is the i^{th} element submatrix of $[\beta]$; $[K_e]_i$ is the i^{th} element stiffness matrix; and $[M_e]_i$ is the i^{th} element mass. The complete formulation is discussed in appendix A.

C. DAMPING MATRIX

The damping forces and hence system damping matrix is not as easily formulated since these values may depend upon the vibration of the system as well as elements exterior to it. To reduce the complexity of the solution of the differential equations of motion linear viscous damping is assumed.

The system damping is formulated by Hurty and Rubinstein, (Ref. 9) by: (1) solving the eigenvalue problem associated with the equations of motion of the system in undamped free vibrations,

for the modal matrix; (2) using this modal matrix to form the diagonalized system mass and stiffness matrices; (3) from these matrices forming the diagonalized system damping matrix; and (4) using this diagonalized damping matrix to solve for the system damping matrix.

The dynamic model consists of two eigenvalue problems. One eigenvalue problem is for the system of Figure 2.5. Here eight coordinates are required to solve the dynamic equations, since the pushrod and rocker arm are separated, coordinate two and coordinate eight have independent motion. The other eigenvalue problem is associated with the model in contact at clearance 2 and takes the form derived by Anderson (2). When the motion at coordinates two and eight are equal, the eight system equations must be reduced to a lower order system by removal of the redundant equation to facilitate a solution. Thus the system damping matrix, as well as the system mass and stiffness matrices, become order seven. This will be considered in section 3.E.

1. Uncoupling the Equations of Motion

When the differential equations of motion are derived from a set of convenient coordinates, the equations are normally coupled. If the system displacements $\{q\}$ are expressed in terms of the normal modes, the governing equations will uncouple. Thus from the equations of free undamped vibration

$$[M]\{\ddot{q}\} + [K]\{q\} = 0 \tag{3.8}$$

the eigenvalue problem may be formulated as

$$\left([\omega^2][M] - [K]\right)\{q\} = \{0\} \quad . \tag{3.9}$$

The solution to this problem will yield the spectral matrix $[\Omega]$ and the modal matrix, $[\Phi]$.

$$[\Omega] = \begin{bmatrix} \omega^2 & & & & 0 \\ & \omega_2^2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ 0 & & & & \omega_n^2 \end{bmatrix} \tag{3.10}$$

$$[\Phi] = \left[\{\Phi\}_1, \{\Phi\}_2, \ldots \{\Phi\}_n\right] \tag{3.11}$$

The integer "n" is the number of degrees of freedom of the system i.e. n = 8 for the first eigenvalue problem discussed above and n = 7 for the second. The $[\omega_i]$ (i = 1, 2, . . . n) are the natural frequencies (or eigenvalues) of the system; and the $\{\Phi\}_i$ (i = 1, 2, ... n) are the column vectors of mode shapes (or eigenvalues) corresponding to the natural frequencies. Now the governing coupled equation of motion for undamped forced vibration,

$$[M]\{\ddot{q}\} + [K]\{q\} = \{F\} \tag{3.12}$$

may be decoupled by performing the coordinate transformation

$$\{q\} = [\Phi]\{\eta\}$$

The dynamic equation can now be written in terms of the modal coordinates $\{\eta\}$,

$$[\diagdown m \diagdown]\{\ddot{\eta}\} + [\diagdown k \diagdown]\{\eta\} = \{F^*\} \tag{3.13}$$

the uncoupled, diagonal mass $\lceil m \rfloor$ and stiffness $\lceil k \rfloor$ matrices, and the modal force vector $\{F^*\}$, Hurty and Rubinstein, (Ref. 9).

2. <u>Relation of System Parameter To Modal Parameter</u>

The diagonal matrices, $\lceil m \rfloor$, $\lceil c \rfloor$ and $\lceil k \rfloor$ are each related to their corresponding system matrices by transformation with the modal matrix.

$$\lceil m \rfloor = [\Phi]^T [M] [\Phi] \tag{3.14}$$

$$\lceil c \rfloor = [\Phi]^T [C] [\Phi] \tag{3.15}$$

$$\lceil k \rfloor = [\Phi]^T [K] [\Phi] \tag{3.16}$$

Additionally the modal acceleration $\{\ddot{\eta}\}$ and displacement $\{\eta\}$ are related to the system acceleration $\{\ddot{q}\}$ and displacement $\{q\}$ by the modal matrix.

$$\{\ddot{q}\} = [\Phi] \{\ddot{\eta}\} \tag{3.17}$$

$$\{q\} = [\Phi] \{\eta\} \tag{3.18}$$

3. <u>Calculation of the System Damping</u>

The system damping matrix may be obtained from equation (3.15) by matrix multiplication. Thus the diagonal elements of $\lceil c \rfloor$ must be calculated before the desired result, $[C]$, can be found. The governing equations are now uncoupled and the uncoupled mass $\lceil m \rfloor$ and stiffness $\lceil k \rfloor$ matrices are known. Thus the solution formulated for the single degree of freedom system may be applied to the n degree of freedom system.

Equations (3.19) through (3.21), from Thomson (Ref. 3), may be utilized to obtain the desired diagonal elements of the uncoupled damping matrix $\left[\diagdown c \diagdown\right]$. The natural frequency in radians per second of the undamped oscillation for a single degree freedom system is

$$\omega_n = \sqrt{k/m} \tag{3.19}$$

The damping factor (ζ), is defined as the ratio of the viscous damping (c) to the critical damping factor (c_c).

$$\zeta = c/c_c \tag{3.20}$$

The critical damping factor is also given by

$$c_c = 2\ m\ \omega_n = 2\sqrt{m\ k} \tag{3.21}$$

Upon substitution of equation (3.21) into equation (3.20), the viscous damping becomes

$$c = 2\,\zeta\sqrt{k\ m} \tag{3.22}$$

In terms of the uncoupled system, the value of the diagonal element of the natural frequencies, corresponding to equation (3.19), are

$$\begin{bmatrix} \omega_1 & & & & 0 \\ & \omega_2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ 0 & & & \cdot & \omega_n \end{bmatrix} = \begin{bmatrix} \frac{k_{11}}{m_{11}} & & & & 0 \\ & \frac{k_{22}}{m_{22}} & & & \\ & & \cdot & & \\ & & & \cdot & \\ 0 & & & \cdot & \frac{k_{nn}}{m_{nn}} \end{bmatrix} \tag{3.23}$$

where the k's and m's are the diagonal elements from $\left[\diagdown k \diagdown\right]$ and $\left[\diagdown m \diagdown\right]$.

Similarly equation (3.22) can be written as

$$c_{ii} = 2\zeta_i \sqrt{k_i \, m_i} \quad (i = 1, 2, \ldots n) \qquad (3.24)$$

and thus the diagonal values of the uncoupled damping matrix $\lceil c \rfloor$ are obtained.

An arbitrary choice of values for the damping factor vector $\{\zeta\}$ was considered on the basis of the natural frequencies. The damping factor, (ζ_i), corresponding to the fundamental frequency is set equal to $\frac{1}{2}$. The remaining elements of $\{\zeta\}$ are set to equal 1/10.

Having established $\lceil c \rfloor$, the damping matrix [C] may be formed from equation (3.15) by matrix multiplication on the left by $[\Phi^T]^{-1}$ and on the right by $[\Phi]^{-1}$ as

$$[\Phi^T]^{-1} \lceil c \rfloor [\Phi]^{-1} = [\Phi^T]^{-1} [\Phi]^T [C] [\Phi] [\Phi]^{-1} \qquad (3.25)$$

This gives the desired result

$$[C] = [\Phi^T]^{-1} \lceil c \rfloor [\Phi]^{-1} \qquad (3.26)$$

D. CONFIGURATION SHIFTING ALGORITHM

For the two clearance problem four configurations were established. This work provided for three clearances having eight possible model configurations. For the addition of the clearance at coordinate four, a total of 4 clearances, there would exist sixteen possible system configurations. Thus the number of possible configurations

of an "n" clearance system would be 2^n. A generalized method of selecting the proper configuration on the basis of the monitored parameters is therefore necessary.

The physical model will shift configurations only on the basis of two general parameters, displacement difference and force. Thus a binary number system facilitates the problem of "computer book-keeping". This is accomplished by assigning the value one to a clearance that is in contact and a zero to a clearance not in contact. For the unique choice of configurations established in chapter 2, Figures 2.4 and 2.5, the configuration matrix is developed as shown in Figure 3.1.

		CLEARANCES		
		#1	#2	#3
CONFIGURATION	1	0	0	0
	2	0	0	1
	3	0	1	0
	4	0	1	1
	5	1	0	0
	6	1	0	1
	7	1	1	0
	8	1	1	1

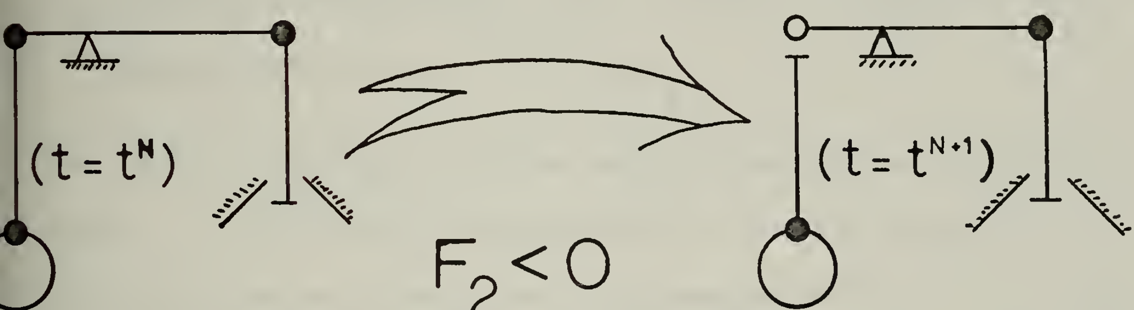
Figure 3.1 Configuration Matrix

The columns of this matrix are associated with clearances 1, 2, and 3 as indicated. The rows are associated with the configuration number that the physical system has taken. Thus to retrieve the correct configuration the entire row is taken as a binary number and converted to a decimal number. Since it is not

convenient to have a configuration numbered zero, a one was arbitrarily added.

For example, when no contact exists at all clearances, the binary number representing this configuration is 0 0 0. This binary number converts to decimal as 0 . Upon adding the above mentioned value one, the decimal value becomes 1, which is the first row of the matrix and hence configuration I.

The binary number 1 1 1 represents contact at all clearances and converts to decimal 7. Adding one yields the value 8, the 8th row of the configuration matrix and hence configuration VIII.



CONFIGURATION VII

CONFIGURATION V

Step 1: VII $\longrightarrow$ 1 1 0

Step 2: $1 \oplus 1 = 1\ 0$

Step 3: 1 0 0

Step 4: 1 0 0 $\longrightarrow$ V

$\oplus$ indicates binary addition

Figure 3.2 Shifting Example No. 1

The manner in which the program determines which configuration the model has shifted into is of primary concern. The developed algorthm is an extension of the above concepts, and best described

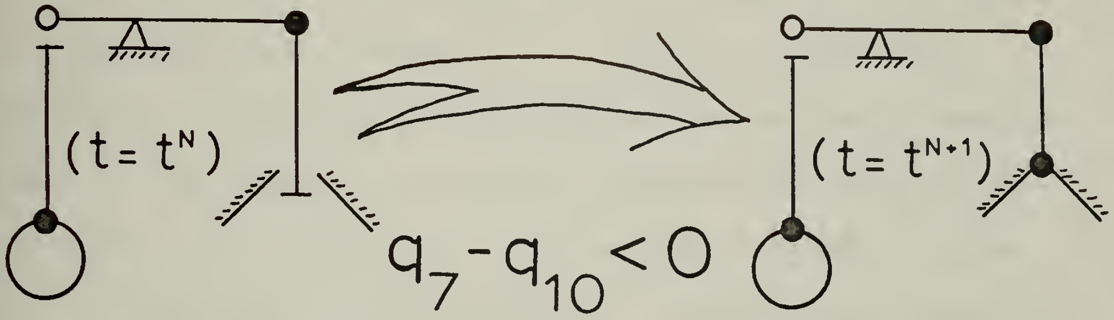
by way of examples. For the dynamic behavior pictured in Figure 3.2, the sequence of events is:

(1) The model is in one of the eight given configurations, i.e. VII at time t^n.

(2) The program calculates the required parameters for the next time step, t^{n+1}. Among these parameters are the monitored values. In this example the monitored value F_2 has changed during the time step, i.e. F_2 has gone from $F_2 > 0$ to $F_2 < 0$.

(3) The negative force at coordinate two indicates the loss of contact at clearance 2. Thus the model now shifts to configuration V for time t^{n+1}, (Figure 2.6).

The four step shifting algorithm accomplishes this as outlined in Figure 3.2. The binary representation for configuration VII is retrieved from row seven in the configuration matrix, 1 1 0. Next the column of the binary number corresponding to the clearance that has been altered is removed. In this example, the element being the second digit of the binary number which correspond to the clearance 2 column. A one is then added, by binary addition, to this element. Carries are disregarded. The new second digit 0 is placed into the binary number. Finally the new resulting binary number is converted to decimal and yields the new configuration number, V.



CONFIGURATION V | CONFIGURATION VI

Step 1: $V \rightarrow 1\ 0\ 0$

Step 2: $0 \oplus 1 = 1$

Step 3: $1\ 0\ 1$

Step 4: $1\ 0\ 1 \rightarrow VI$

$\oplus$ indicates binary addition

Figure 3.3 Shifting Example No. 2

The second example for the shifting algorithm is shown in Figure 3.3. The sequence of events for the dynamic behavior is:

(1) The model is in configuration V at time t^n.

(2) The program calculates the required parameters at the next time step t^{n+1}. In this example, the monitored displacement difference value $q_7 - q_{10}$ has changed from $q_7 - q_{10} > 0$ to $q_7 - q_{10} < 0$.

(3) The negative displacement difference at the coordinate pair 7 - 10 indicates that contact is established at clearance 3. Thus the model now shifts to configuration VI for time t^{n+1}.

The shifting algorithm accomplishes this as outlined in Figure 3.3. The binary representation for configuration V is retrieved from row five in the configuration matrix, 1 0 0. Next the column of the binary number corresponding to the clearance that has been altered is removed. In this example, the element being the third digit of the binary number which corresponds to the clearance 3 column. The new digit, 1 is placed into the binary number. Finally the new resulting binary number is converted to decimal and yields the new configuration number, VI.

E. METHOD OF CALCULATIONS

Newmark's β parameter method, (Ref. 4), forms the basis for the calculations. This one step numerical intergration procedure assumes the accelerations of the system coordinates are a linear function of time. Errors introduced by this assumption are given by $\omega_n \Delta T/6$, where ω_n is the highest natural frequency of interest. For small time steps this error is negligible. In this study the time step is taken as

$$\Delta T = 1 \times 10^{-6} \text{ seconds} \tag{3.27}$$

It will be shown, by example that the solution to the equations of motion takes a convenient form. Precalculated matrix expressions may generally be used, resulting in large savings of computer execution time. A basic familiarity with the formulation of Newmark's method (Appendix B) will prove helpful and aid in the understanding of the following discussion.

The development of the problem calculations will assume the model has been operating in configuration VI, contact at the cam and valve seat. The procedures are identical by extension, to the remaining seven configurations. Thus the problem unknown quantities have been calculated for time $t = t^n$. The object is to solve the governing equations for the next time increment $t = t^{n+1}$.

1. Solution to the Governing Equations

The full system matrices [M], [C], and [K] have been developed in sections 3.B and 3.C. The governing equations of motion are expressed as the matrix equation 3.1 in section 3.A. Written in concise form the governing equations for the eight system coordinates are

$$\left[m_{ij}\right] \{\ddot{q}_i\}^{n+1} + \left[c_{ij}\right] \{\dot{q}_i\}^{n+1} + \left[k_{ij}\right] \{q_i\}^{n+1} = \{F_i\}^{n+1}$$

$$(i = 1,2,\ldots 8;\ j = 1,2,\ldots 8) \qquad (3.28)$$

Exterior forces, F_1^{n+1} and F_7^{n+1}, are unknown. The remaining applied forces F_i^{n+1} $(i = 2,3,..6,8)$ are equal to zero. Since the cam and follower are in contact, the known cam values at reference coordinate nine, $\ddot{q}_9^{n+1}$, $\dot{q}_9^{n+1}$, and q_9^{n+1} are equal to the corresponding follower values at system coordinate one, $\ddot{q}_1^{n+1}$, $\dot{q}_1^{n+1}$, and q_1^{n+1}. Similiarly, the known values for the stationary valve seat at reference coordinate ten, $\ddot{q}_{10}^{n+1}$, $\dot{q}_{10}^{n+1}$ and q_{10}^{n+1} are set equal to those corresponding values for the valve at system coordinate seven, $\ddot{q}_7^{n+1}$, $\dot{q}_7^{n+1}$, and q_7^{n+1}. Displacements

q_i^{n+1} $(i = 2, 3, \ldots 6, 8)$ are unknown. Expanding and partitioning equations 3.28 to take advantage of the known values yields

$$
\begin{bmatrix}
m_{11} & m_{12} & \cdots & m_{16} & m_{17} & m_{18} \\
m_{21} & m_{22} & \cdots & m_{26} & m_{27} & m_{28} \\
\cdot & & & & & \\
\cdot & & & & & \\
\cdot & & & & & \\
m_{61} & m_{62} & \cdots & m_{66} & m_{67} & m_{68} \\
m_{71} & m_{72} & \cdots & m_{76} & m_{77} & m_{78} \\
m_{81} & m_{82} & \cdots & m_{86} & m_{87} & m_{88}
\end{bmatrix}
\begin{Bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \cdot \\ \cdot \\ \cdot \\ \ddot{q}_6 \\ \ddot{q}_7 \\ \ddot{q}_8 \end{Bmatrix}^{n+1}
+
\begin{bmatrix}
c_{11} & c_{12} & \cdots & c_{16} & c_{17} & c_{18} \\
c_{21} & c_{22} & \cdots & c_{26} & c_{27} & c_{28} \\
\cdot & & & & & \\
\cdot & & & & & \\
\cdot & & & & & \\
c_{61} & c_{62} & \cdots & c_{66} & c_{67} & c_{68} \\
c_{71} & c_{72} & \cdots & c_{76} & c_{77} & c_{78} \\
c_{81} & c_{82} & \cdots & c_{86} & c_{87} & c_{88}
\end{bmatrix}
\begin{Bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \cdot \\ \cdot \\ \cdot \\ \dot{q}_6 \\ \dot{q}_7 \\ \dot{q}_8 \end{Bmatrix}^{n+1}
$$

$$
+
\begin{bmatrix}
k_{11} & k_{12} & \cdots & k_{16} & k_{17} & k_{18} \\
k_{21} & k_{22} & \cdots & k_{26} & k_{27} & k_{28} \\
\cdot & \cdot & \cdots & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdots & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdots & \cdot & \cdot & \cdot \\
k_{61} & k_{62} & \cdots & k_{66} & k_{67} & k_{68} \\
k_{71} & k_{72} & \cdots & k_{76} & k_{77} & k_{78} \\
k_{81} & k_{82} & \cdots & k_{86} & k_{87} & k_{88}
\end{bmatrix}
\begin{Bmatrix} q_1 \\ q_2 \\ \cdot \\ \cdot \\ \cdot \\ q_6 \\ q_7 \\ q_8 \end{Bmatrix}^{n+1}
=
\begin{Bmatrix} F_1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ F_7 \\ 0 \end{Bmatrix}^{n+1}
\qquad (3.29)
$$

The next step creates two sets of equations. The first set, equations 3.30 and 3.31 consists of individual equations in which the displacement, velocity,and accelerations are known but the force is not. The second set, equation 3.32 consists of the matrix equations set equal to the zero force vector. In these equations all displacement, velocity,and acceleration quantities are unknown.

These equations of motion for the system are now written as:

$$F_1 = \langle m_{11} \cdots m_{18} \rangle \begin{Bmatrix} \ddot{q}_1 \\ \cdot \\ \cdot \\ \cdot \\ \ddot{q}_8 \end{Bmatrix}^{n+1} + \langle c_{11} \cdots c_{18} \rangle \begin{Bmatrix} \dot{q}_1 \\ \cdot \\ \cdot \\ \cdot \\ \dot{q}_8 \end{Bmatrix}^{n+1} + \langle k_{11} \cdots k_{18} \rangle \begin{Bmatrix} q_1 \\ \cdot \\ \cdot \\ \cdot \\ q_8 \end{Bmatrix}^{n+1} \tag{3.30}$$

$$F_7 = \langle m_{71} \cdots m_{78} \rangle \begin{Bmatrix} \ddot{q}_1 \\ \cdot \\ \cdot \\ \cdot \\ \ddot{q}_8 \end{Bmatrix}^{n+1} + \langle c_{71} \cdots c_{78} \rangle \begin{Bmatrix} \dot{q}_1 \\ \cdot \\ \cdot \\ \cdot \\ \dot{q}_8 \end{Bmatrix}^{n+1} + \langle k_{71} \cdots k_{78} \rangle \begin{Bmatrix} q_1 \\ \cdot \\ \cdot \\ \cdot \\ q_8 \end{Bmatrix}^{n+1} \tag{3.31}$$

$$\begin{bmatrix} m_{22} & m_{23} & \cdots & m_{26} & m_{28} \\ m_{32} & m_{33} & \cdots & m_{36} & m_{38} \\ \vdots & \vdots & & \vdots & \vdots \\ m_{62} & m_{63} & \cdots & m_{66} & m_{68} \\ m_{82} & m_{83} & \cdots & m_{86} & m_{88} \end{bmatrix} \begin{Bmatrix} \ddot{q}_2 \\ \ddot{q}_3 \\ \vdots \\ \ddot{q}_6 \\ \ddot{q}_8 \end{Bmatrix}^{n+1} + \begin{bmatrix} c_{22} & c_{23} & \cdots & c_{26} & c_{28} \\ c_{32} & c_{33} & \cdots & c_{36} & c_{38} \\ \vdots & \vdots & & \vdots & \vdots \\ c_{62} & c_{63} & \cdots & c_{66} & c_{68} \\ c_{82} & c_{83} & \cdots & c_{86} & c_{88} \end{bmatrix} \begin{Bmatrix} \dot{q}_2 \\ \dot{q}_3 \\ \vdots \\ \dot{q}_6 \\ \dot{q}_8 \end{Bmatrix}^{n+1} + \begin{bmatrix} k_{22} & k_{23} & \cdots & k_{26} & k_{28} \\ k_{32} & k_{33} & \cdots & k_{36} & k_{38} \\ \vdots & \vdots & & \vdots & \vdots \\ k_{62} & k_{63} & \cdots & k_{66} & k_{68} \\ k_{82} & k_{83} & \cdots & k_{86} & k_{88} \end{bmatrix} \begin{Bmatrix} q_2 \\ q_3 \\ \vdots \\ q_6 \\ q_8 \end{Bmatrix}^{n+1}$$

$$+ \Delta P_1^{\,n+1} + \Delta P_2^{\,n+1} = \begin{Bmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{Bmatrix} \tag{3.32}$$

Where

$$\Delta P_1^{n+1} = \begin{Bmatrix} m_{21} \\ m_{31} \\ \cdot \\ \cdot \\ m_{61} \\ m_{81} \end{Bmatrix} \ddot{q}_1^{n+1} + \begin{Bmatrix} c_{21} \\ c_{31} \\ \cdot \\ \cdot \\ c_{61} \\ c_{81} \end{Bmatrix} \dot{q}_1^{n+1} + \begin{Bmatrix} k_{21} \\ k_{31} \\ \cdot \\ \cdot \\ k_{61} \\ k_{81} \end{Bmatrix} q_1^{n+1} \tag{3.33}$$

and

$$\Delta P_2^{n+1} = \begin{Bmatrix} m_{27} \\ m_{37} \\ \cdot \\ \cdot \\ m_{67} \\ m_{87} \end{Bmatrix} \ddot{q}_7^{n+1} + \begin{Bmatrix} c_{27} \\ c_{37} \\ \cdot \\ \cdot \\ c_{67} \\ c_{87} \end{Bmatrix} \dot{q}_7^{n+1} + \begin{Bmatrix} k_{27} \\ k_{37} \\ \cdot \\ \cdot \\ k_{67} \\ k_{87} \end{Bmatrix} q_7^{n+1} \tag{3.34}$$

The set of consistent system equations written in concise form are

$$F_1 = \sum_{i=1}^{8} (m_{1i}\, \ddot{q}_i^{n+1} + c_{1i}\, \dot{q}^{n+1} + k_{1i}\, q^{n+1}) \tag{3.35}$$

$$F_7 = \sum_{i=1}^{8} (m_{7i}\, \ddot{q}_i^{n+1} + c_{7i}\, \dot{q}^{n+1} + k_{7i}\, q^{n+1}) \tag{3.36}$$

and

$$[m_{ij}]\, \{\ddot{q}_i\}^{n+1} + [c_{ij}]\, \{\dot{q}_i\}^{n+1} + [k_{ij}]\, \{q_i\}^{n+1} + \sum_{j=1}^{2} \Delta P_j^{n+1} = \{0_i\} \tag{3.37}$$

$$(i = 2,3,..6,8;\ j = 2,3,..6,8)$$

The next step is the solution of the matrix equation 3.37 by Newmark's method, for the unknown accelerations. By integration the velocities and displacements are written in terms of the accelerations as:

$$\{\dot{q}_i\}^{n+1} = \{\dot{q}_i\}^n + \frac{\Delta T}{2}\{\ddot{q}_i\}^n + \frac{\Delta T}{2}\{\ddot{q}_i\}^{n+1} \tag{3.38}$$

and

$$\{q_i\}^{n+1} = \{q_i\}^n + \Delta T\{\dot{q}_i\}^n + \frac{\Delta T^2}{3}\{\ddot{q}_i\}^n + \frac{\Delta T^2}{6}\{\ddot{q}_i\}^{n+1} \tag{3.39}$$

$$(i = 2,3, .. 6,8)$$

Let

$$\{\alpha_i\}^n = \{\dot{q}_i\}^n + \frac{\Delta T}{2}\{\ddot{q}_i\}^n \tag{3.40}$$

and

$$\{\beta_i\}^n = \{q_i\}^n + \Delta T\{\dot{q}_i\}^n + \frac{\Delta T^2}{3}\{\ddot{q}_i\}^n \tag{3.41}$$

$$(i = 2,3, .. 6,8)$$

Substituting for $\{\alpha\}^n$ and $\{\beta\}^n$ equations 3.38 and 3.39 become:

$$\{\dot{q}_i\}^{n+1} = \{\alpha_i\}^n + \frac{\Delta T}{2}\{\ddot{q}_i\}^{n+1} \tag{3.42}$$

$$\{q_i\}^{n+1} = \{\beta_i\}^n + \frac{\Delta T^2}{6}\{\ddot{q}_i\}^{n+1} \tag{3.43}$$

$$(i = 2,3,..6,8)$$

Next substituting for $\{\dot{q}_i\}^{n+1}$ and $\{q_i\}^{n+1}$ into equations 3.37 and factoring yields:

$$\left([m_{ij}] + [c_{ij}]\frac{\Delta T}{2} + [k_{ij}]\frac{\Delta T^2}{6}\right)\{\ddot{q}_i\}^{n+1}$$

$$+ [c_{ij}]\{\alpha_i\}^n + [k_{ij}]\{\beta_i\}^n + \sum_{k=1}^{2} \Delta P_k^{\,n+1} = \{0_i\} \qquad (3.44)$$

$$(i = 2,3 \,..\, 6,8;\ j = 2,3,\ ..\ 6,8)$$

Let

$$[BMIV_{ij}]^{-1} = [m_{ij}] + [c_{ij}]\frac{\Delta T}{2} + [k_{ij}]\frac{\Delta T^2}{6} \qquad (3.45)$$

$$(i = 2,3,\ ..\ 6,8;\ j = 2,3,\ ..\ 6,8)$$

Substituting for $[BMIV]^{-1}$ in equation 3.44 and moving the knowns to the right side of the equation yields

$$[BMIV_{ij}]^{-1}\{\ddot{q}_i\}^{n+1} = -[c_{ij}]\{\alpha_i\}^n - [k_{ij}]\{\beta_i\}^n - \sum_{k=1}^{2} \Delta P_k^{\,n+1} \qquad (3.46)$$

$$(i = 2,3,..6,8\ ;\ j = 2,3,..6,8)$$

Inverting the matrix $[BMIV]^{-1}$ yields the expression for the unknown accelerations

$$\{\ddot{q}_i\}^{n+1} = -[BMIV_{ij}][c_{ij}]\{\alpha_i\}^n - [BMIV_{ij}][k_{ij}]\{\beta_i\}^n - [BMIV_{ij}]\left(\sum_{k=1}^{2} \Delta P_k^{\,n+1}\right)$$

$$(i = 2,3,..6,8;\ j = 2,3,..6,8) \qquad (3.47)$$

In equation 3.47 the non-time function matrix multiplications operations are precalculated as

$$[DA_{ij}] = [BMIV_{ij}]\,[c_{ij}] \qquad (3.48)$$

and

$$[DV_{ij}] = [BMIV_{ij}]\,[k_{ij}] \qquad (3.49)$$

$$(i = 2,3,..6,8;\ j = 2,3,..6,8)$$

The system accelerations, except the initial known values, are now calculated by

$$\{\ddot{q}_i\}^{n+1} = - [DA_{ij}] - [DV_{ij}] - [BMIV_{ij}] \left(\sum_{k=1}^{2} \Delta P_k^{\ n+1}\right) \qquad (3.50)$$

$$(i = 2,3,..6,8;\ j = 2,3,..6,8)$$

The system velocities and displacements are calculated from

$$\{\dot{q}_i\}^{n+1} = \{\alpha_i\} + \frac{\Delta T}{2} \{\ddot{q}_i\}^{n+1} \qquad (3.51)$$

$$\{q_i\}^{n+1} = \{\beta_i\} + \frac{\Delta T^2}{6} \{\ddot{q}_i\}^{n+1} \qquad (3.52)$$

$$(i = 2,3,...6,8;\ j = 2,3,...6,8)$$

The remaining system unknowns, F_1 and F_7 may be directly calculated from equations 3.30 or 3.35 and 3.31 or 3.36. This completes the calculations for the required solution to the problem for this time step.

2. Precalculated Matrices

Returning to equation 3.32 it is noted that the mass, damping, and stiffness matrices are unique to configuration VI. Similarly from equation 3.30 and 3.31 the mass, damping, and stiffness column vectors are also unique to configuration VI. Thus these matrices need only be calculated once if stored as layers to the full system matrices. Figure 3.4 graphically indicates the matrix dimensions and elements for this and the remaining seven configurations. Additionally for the eight sets of precalculated matrices [M], [C], and [K] the matrix variables [BMIV], [DA], and [DV] given by equations 3.45, 3.48, and 3.49 may be precalculated for the standard ΔT and stored.

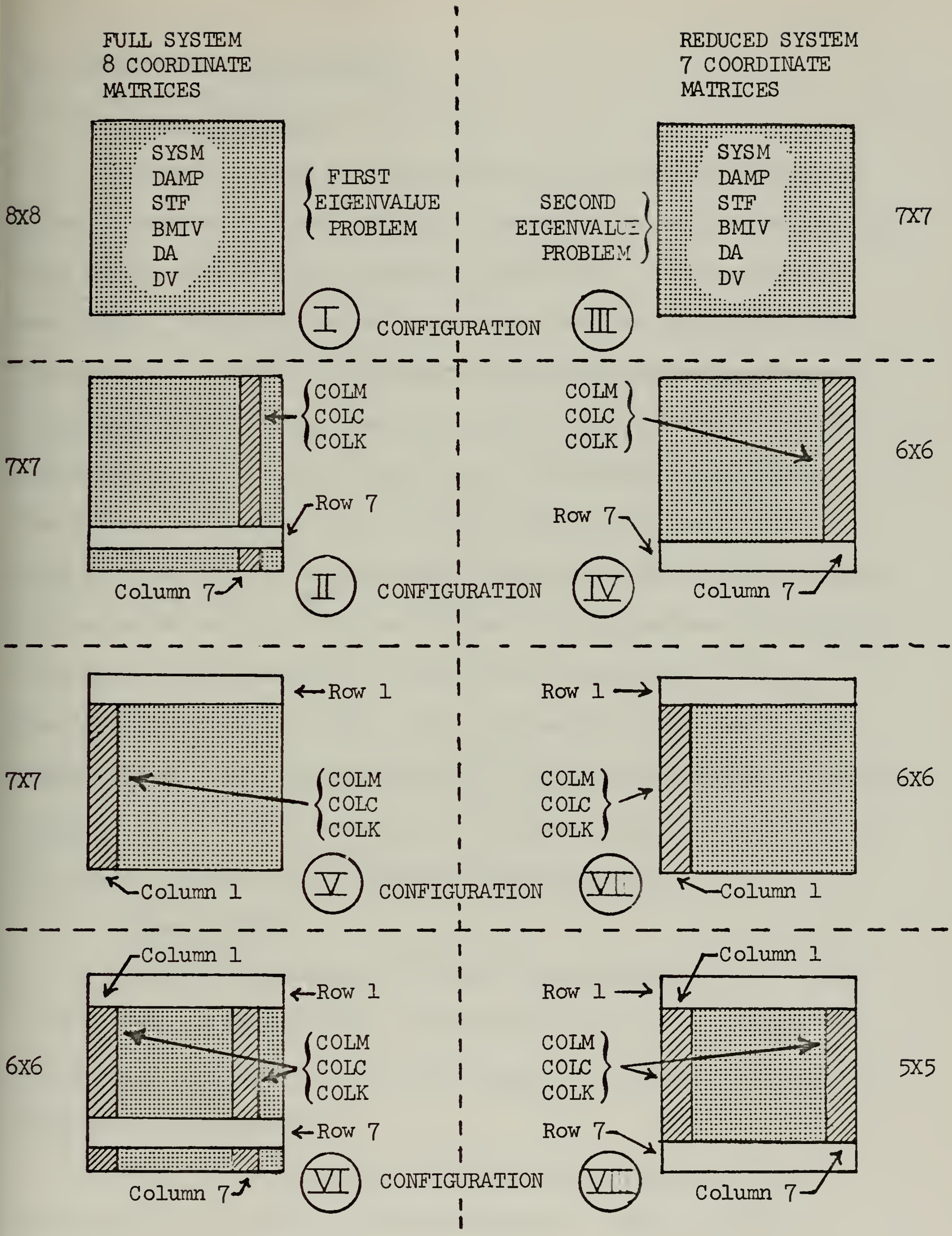


Figure 3.4 MATRIX LAYERS

F. REFINEMENTS IN THE SOLUTION

1. Incremental Time

As seen in section 3.E, equation 3.27, the normal time step is established as 1×10^{-6} seconds. Generally the system configuration will change in a time step interval, rather than at either of the end points. When this occurs the present negative monitored value calculated at $t = t^{n+1}$ and the past stored monitored value that was calculated at $t = t^{n}$ are used for a linear interpolation to obtain the zero point, i.e. zero force or zero displacement difference. The ΔT, now designated as $\Delta T'$, corresponding to the zero point is used to calculate the system values based on the old configurations. If the system has more than one change in configurations for a given time step, the minimum $\Delta T'$ is used for the calculations, $\Delta T'_{min}$. The newly calculated critical value is then monitored, and if still negative, the time is again reduced by a small amount until a positive value of the monitored parameter is obtained. Figure 3.5 summerizes the above procedures for a negative force parameter.

2. Impact Forces

Anderson, in Chapter IV (Ref. 2), puts forth a good account of the study of impact forces. Numerous references are cited. Thus the historical background of impact forces in relation to this study will not be discussed here.

When the displacement difference values reduce to zero (or negative), contact at a clearance exists. This produces a step input to the velocity function and an impulse to the acceleration

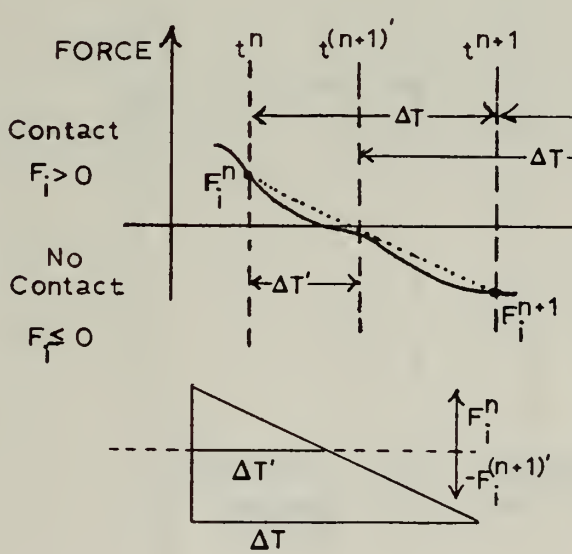
function. Finite time steps are utilized to obtain solutions. Therefore one time step is sufficient to reach the step input value for the velocity curve, but two time intervals are required to account for the acceleration caused by the impulse, Figure 3.6. Thus the acceleration at time t^{n+1} is

$$\ddot{q}_i^{\,n+1} = (\dot{q}_i^{\,n+1} - \dot{q}_i^{\,n}) / \Delta T \qquad (i = 1,2,7) \tag{3.53}$$

This accelation is readily calculated for impact at the cam or valve seat (i = 1 or 7). However for contact at the rocker, neither $\ddot{q}_2^{\,n+1}$ or $\dot{q}_2^{\,n+1}$ is known. In order to solve the equation the assumption is made that

$$\dot{q}_2^{\,n+1} = \text{SCALE} * \dot{q}_2^{\,n} \quad . \tag{3.54}$$

The value for SCALE is arbitrarily chosen from 1.0 to 2.0, until a good system response results. Further study is suggested in this area.



$$\frac{\Delta T'}{\Delta T} = \frac{F_i^n}{F^n - F^{(n+1)'}}$$

$$\Delta T' = \left(\frac{F_i^n}{F_i^n - F^{(n+1)'}} \right) \Delta T$$

then $\quad \omega t^{(n+1)'} = \omega t^n + \omega \Delta T'$

is used to calculate system values.

Figure 3.5 Incremental Time $\Delta T'$

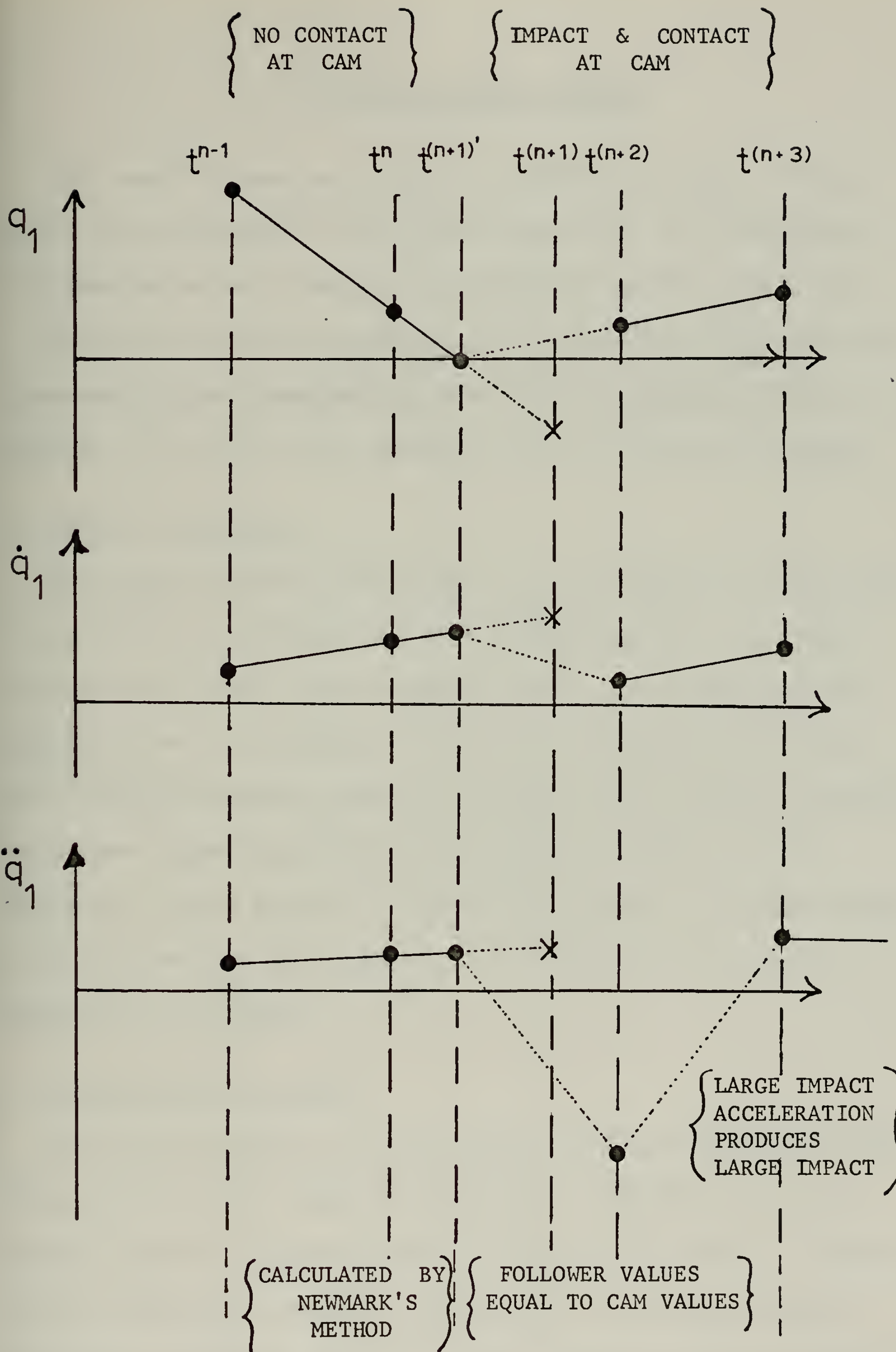


Figure 3.6 System Curves For Impact Considerations

IV. IBM-360 COMPUTER PROGRAM

This computer program is coded in FORTRAN IV, (Refs. 10-13), to run on an IBM-360/67, with OS/360 release 18. The simulation is modeled after basic concepts put forth by Anderson, (Ref. 2).

The original model program has been completely restructured and formulated to yield general procedures for the analysis of the dynamics of an elastic link mechanism with an internal clearance.

A. GENERAL DESCRIPTION

The program requires 234 K bytes of core storage for 850 degrees of cam rotation, including the CALCOMP graph buffer. The offline printer array (GRID) is not included in this core storage figure. Execution time is a function of the speed of rotation of the cam. For 11,000 RPM with 850 degrees, execution time is about 4.3 minutes. The primary output parameters fill arrays for SYSOUT = B and SYSOUT = C. The B queue is for transfering data, via punched cards, for use in the SDS-9300 computer. The C queue is for plotting graphs via the CALCOMP.

B. GENERALIZATION MATRICES

As noted in section 3.D, the number of system configurations is 2^n, where "n" is the number of clearances. Therefore, the most general computer programming was utilized, where possible. Several arrays of prestored computer "instructions" are either computer generated or formed by the user. These arrays set looping variables,

dynamically dimension work arrays, and establish program calculation flow indicators.

1. Matrix CONFIG

The 8 by 3 matrix CONFIG, Figure 4.1 was described in section 3.D. This integer array indicates the physical configurations that the model has taken based on the value of the monitored parameters. The matrix is formed by the program. The user must supply the value for the maximum number of configurations the system is permitted to take.

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Figure 4.1 Matrix CONFIG

2. Matrix IDM

The 8 by 1 column vector IDM indicates the dimension of the matrices that are used for calculations. This array is indicated in Figure 4.2 for the three clearance system. The program parameter ISYS is the matrix dimension of the equivalent system mass, stiffness, and damping matrices and for the calculated program matrix variables BMIV, DA, and DV. The dimensions of the various layers correspond to their configuration number and to the rows of IDIM, i.e. when the system is in configuration I, layer one matrices have the dimension indicated in row one of IDIM. This integer array is calculated by the program.

$$\left\{ \begin{matrix} 8 \\ 7 \\ 7 \\ 6 \\ 7 \\ 6 \\ 6 \\ 5 \end{matrix} \right\}$$

Figure 4.2 Matrix IDIM

3. Matrix LAYER

The 8 by 1 array LAYER, Figure 4.3, is an integer column vector supplied by the user. The elements of the vector consist of ones or threes. As the name implies, the layer of the mass, stiffness, damping, BMIV, DA, and DV matrices used in the calculations correspond to the configuration number the system has taken. The eigenvalue

problem for contact at the internal clearance (layer 1) or no contact (layer 3), determines the elements of LAYER. Configurations III, IV, VII , or VIII for contact at the rocker-pushrod, use the same eigenvalue problem for the reduced system, i.e. the damping matrix is calculated with layer 3 matrices. Configurations I, II, V, or VI for no contact at the rocker-pushrod, use the full system eigenvalue problem. The damping matrix is calculated using layer 1 matrices. Figure 4.3 shows these values of LAYER for the three clearance model with two eigenvalue problems.

$$\begin{Bmatrix} 1 \\ 1 \\ 3 \\ 3 \\ 1 \\ 1 \\ 3 \\ 3 \end{Bmatrix}$$

Figure 4.3 Matrix LAYER

4. Matrix IQ

The 8 by 8 integer array IQ, Figure 4.4 indicates the system coordinates for which the Newmark routine will calculate the unknown quantities i.e. displacements, velocities, accelerations, and forces for a given time step. The columns of the array correspond to the configuration number except column 1 and 3. These are additional columns used with configurations VI and VIII respectively. The array is user supplied.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 7 & 8 & 7 & 0 \\ 7 & 8 & 7 & 0 & 8 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 4.4 Matrix IQ

5. Matrix IDP

This 8 by 3 integer array, Figure 4.5, consists of zero elements except where impact may be expected to occur at the external clearances. Here again the rows correspond to the configuration number and the columns correspond to the clearance number. The matrix IDP is used in the calculation of ΔP as a portion of Newmark's method. The array is calculated by the program.

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 0 & 0 & 0 \\ 0 & 0 & 7 \\ 1 & 0 & 0 \\ 1 & 0 & 7 \\ 1 & 0 & 0 \\ 1 & 0 & 7 \end{bmatrix}$$

Figure 4.5 Matrix IDP

6. Matrix ICORD

The 3 by 2 integer array ICORD, Figure 4.6 indicates the coordinate pairs of the internal and external clearances. The elements are stored row wise by pairs. The rows correspond to the clearance numbers 1, 2, and 3 respectively. The first column corresponds to the index of the system coordinate that has the greater positive displacement when no contact exists at the clearance in question. The second column is assigned to the remaining index of the clearance pair. The array ICORD is user generated. Figure 4.6 shows the matrix for the three clearance problem defined in chapter 2.

$$\begin{bmatrix} 1 & 9 \\ 2 & 8 \\ 7 & 10 \end{bmatrix}$$

Figure 4.6 Matrix ICORD

7. Matrix IMPACT

The 3 by 1 column vector IMPACT is program generated. The integer elements are continuously modified during program execution. The rows correspond to the clearance number pairs. Values 1, 2, or 7, of the system coordinates are stored when impact occurs for a given time step at a corresponding clearance.

C. DYNAMIC DIMENSIONING

The principle of dynamic dimensioning is widely used in advanced programming and often misunderstood. Thus a short example of its concepts as applied to this work is presented here. Reference to Figures 4.7 and 4.8 will aid in the understanding of the concepts.

ARRAY SYSM

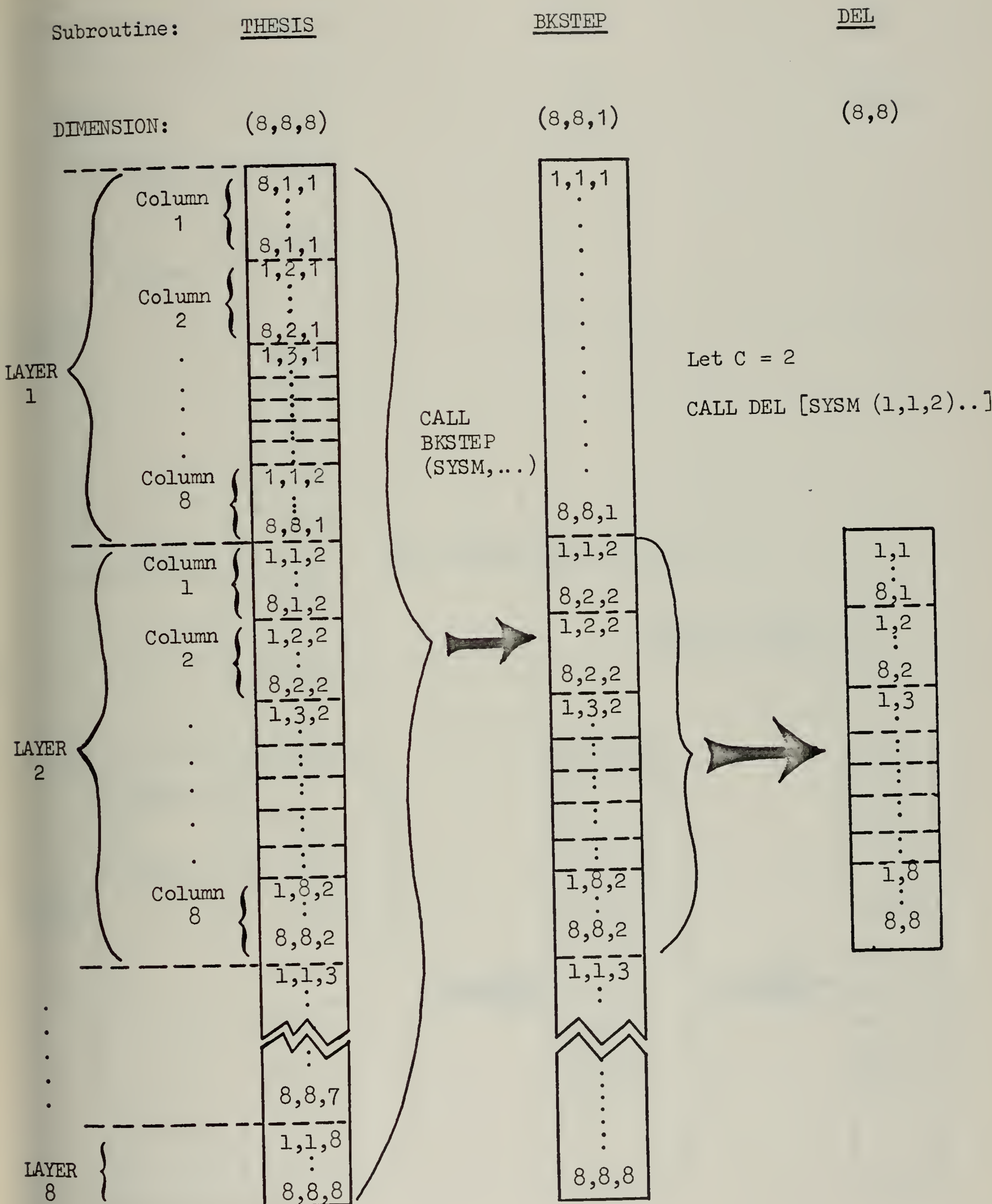


Figure 4.7 Dynamic Dimensioning of Array SYSM

ARRAY BMIV

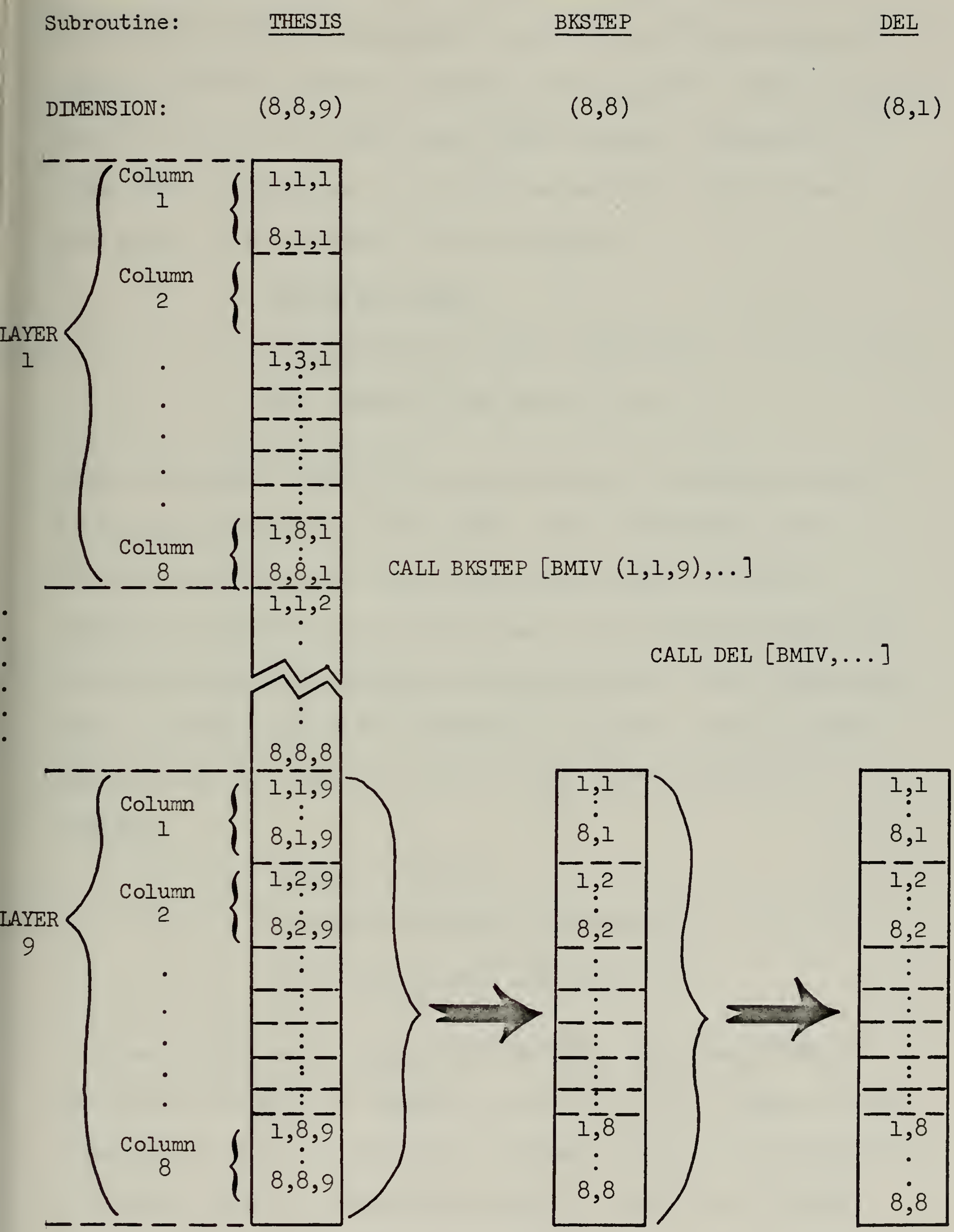


Figure 4.8 Dynamic Dimensioning of Array BMIV

In subroutine THESIS the mass array SYSM and the calculation array BMIV are exactly dimensioned indicating the total number of elements which are stored by columns in the computer. When subroutine BKSTEP is called, the entire array SYSM is passed. However the array BMIV is initialized in the call as the first value of the last layer, in this example. This is shown as:

```
Subroutine THESIS
DIMENSION SYSM (8,8,8), BMIV(8,8,9)
CALL  BKSTEP (SYSM, BMIV(1,1,9), ...)
```

Thus in subroutine BKSTEP, the array SYSM must be dimensioned with the exact number of rows and columns. Since the computer has already provided for the total space of the array, the third dimension is dynamic and if entered may be any positive number equal to or less than the originally assigned dimension. Since the entire array is utilized the third dimension is retained. For the array BMIV only the last layer is used, thus the third dimension is dropped.

```
Subroutine BKSTEP
DIMENSION  SYSM(8,8,1), BMIV(8,8)
CALL  DEL (SYSM(1,1,C), BMIV, ...)
```

When subroutine DEL is called from BKSTEP the array, SYSM is now initialized with the variable C, (C = 1,2,...8). BMIV, now only considered an 8 by 8 array, is not initialized here. The dimensioning in subroutine DEL is accomplished similar to that done in BKSTEP.

The array SYSM must again be dimensioned as above with the third dimensioning parameter being dynamic. The array BMIV is used as a double dimensioned array in subroutine DEL. By arguments similar to those above, the columns of the array may now also be dynamically dimensioned as:

```
Subroutine DEL
DIMENSION SYSM(8,8), BMIV(8,1)
```

Figure 4.7 for array SYSM and Figure 4.8 for array BMIV graphically indicate the manner in which these arrays are passed from the subroutines, THESIS-BKSTEP-DEL.

D. PROGRAM STRUCTURE

The main program only sets up the plotting arrays and then calls subroutine THESIS, the main routine that controls the program execution. The calling order is presented in Figure 4.9. The CALCOMP graphing subroutines indicated by an asterisk are supplied by the computer center.

E. SUBROUTINES

The function of each of the subroutines and a description of their execution are included in this section. The concept of execution of object modules stored on disk was used to a large extent. This accounts for the large number of subroutines and their limited content. Flow graphs for the majority of the routines are collected and presented at the end of this section. Each subroutine is listed in its entirety in the IBM-360 Computer Program Listing chapter.

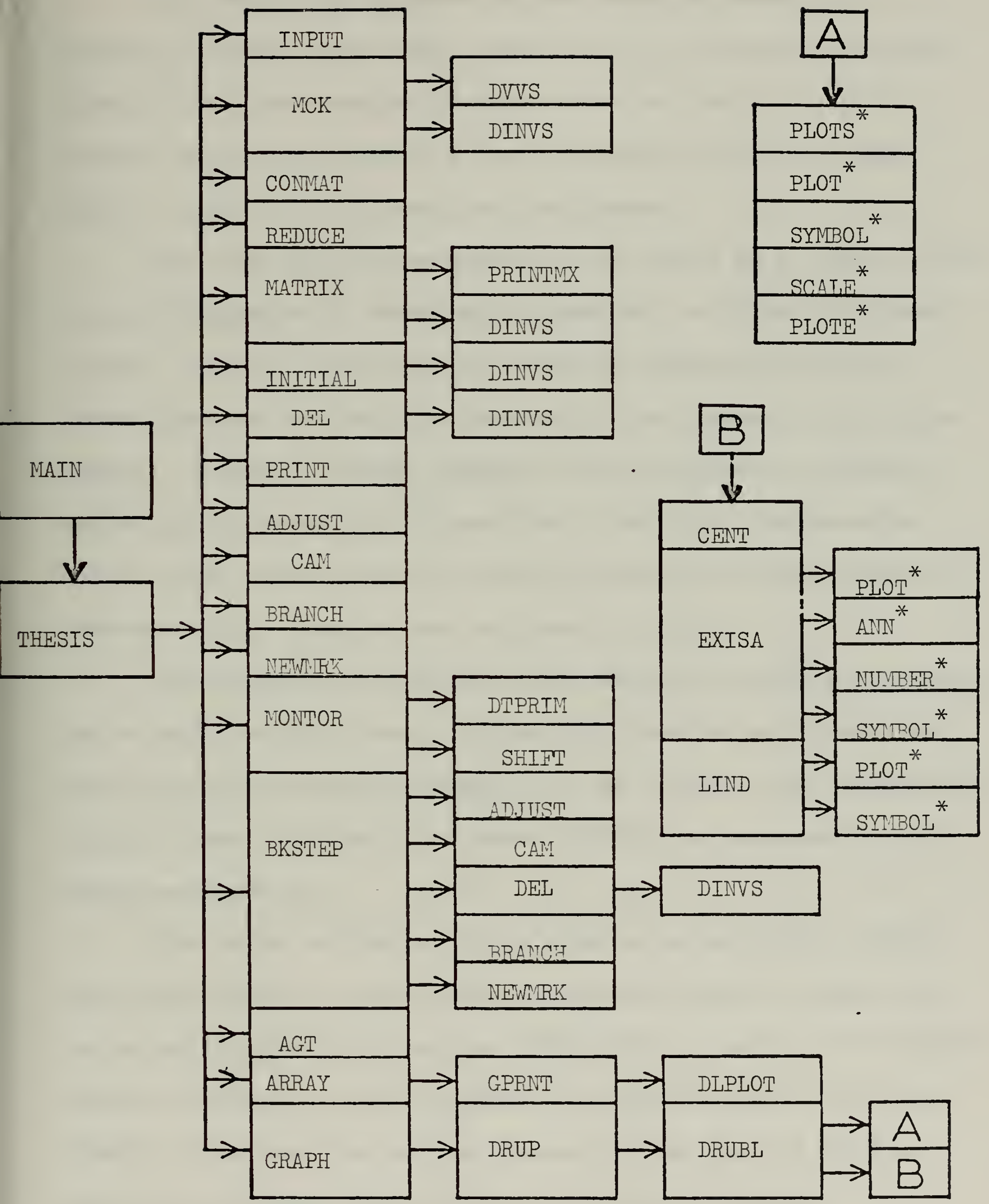


Figure 4.9 IBM-360 Computer Program Structure

1. <u>Subroutine THESIS</u>

This subroutine controls program execution, establishes calculation arrays (MCK, CONMAT, MATRIX, DEL), initialing parameters (INPUT, INITAL) and contains the major time step loop, $\Delta T = 10^{-6}$ seconds. As seen in Figure 4.9, most subroutines are called from THESIS. Figure 4.10 indicates the flow diagram.

The time parameter ωt advances by an amount $\omega \Delta T$. The previous values of displacement, velocity, acceleration, and force are stored (ADJUST). The new cam values are calculated (CAM) and the known system parameter are initialed according to the present configuration (BRANCH). Newmark's method (NEWMARK) is then employed to calculate the remaining unknown system quantities. The system configuration value is next stored prior to checking the newly calculated quantities (MONTOR) for any change in the configuration number.

The system is now checked to see whether an impact has occurred and at what coordinate index. If the configuration has changed, the quantities are recalculated (BKSTEP) for the time that the configuration changed. Then force values are reset (FIXFOR) to coorespond to the system configuration.

The amount of degrees of cam rotation is calculated. If the next whole degree has been reached the system values are output for use as the graphics (AGT) and draw (ARRAY) data. Next if the required number of degrees have been evaluated the desired graphs are plotted (GRAPH). Otherwise, the program branches to the start of the time step loop (statement number 360) and the cycle is repeated.

2. Subroutine BKSTEP

This subroutine is similar to subroutine THESIS in that all program variables are recalculated to correspond to the time at which the system configuration changed. Once this routine is entered only the parameter corresponding to the configuration shift is monitored until a positive value is calculated. Figure 4.11 indicates the flow diagram.

3. Subroutine MCK, DVVS, and DINVS

This subroutine originally coded by Anderson and Winfrey, was modified to account for the additional eigenvalue problem. The sizes and material properties of the system members are assigned. The element of the full and reduced system mass and stiffness are assigned to their respective arrays, SYSM and STF. The damping ratio, (ZTA) is set on the basis of the fundamental frequency of the system being considered. The eigenvalue problems are solved by subroutine DVVS. Subroutine DINVS is called to obtain matrix inverses..

4. Newmark's Method Subroutines

Subroutines NEWMRK and DEL are the primary program calculation routines, and considered together comprise Newmark's method. The flow diagram for subroutine DEL is seen in Figure 4.12 and for subroutine NEWMRK, Figure 4.13.

(a) Subroutine DEL

Subroutine DEL calculates the eight layers of the matrix arrays BMIV, DA, and DV, based on the full time step ΔT. These matrices are precalculated and stored. When DEL is called from subroutine BKSTEP, an additional layer of the above matrices is calculated as required, with a partial time step, $\Delta T_{min} < 10^{-6}$ seconds.

(b) Subroutine NEWMRK

Subroutine NEWMRK calculates the coordinate accelerations by Newmark's method. The theory of this method is presented in appendix B. Its relation to this study is presented in section 3.E. After the system accelerations are known, the displacements and velocities can be calculated. Finally the forces are calculated at clearance coordinates 1, 2, and 7.

5. Subroutine for Shifting Configurations

The two subroutines that provide the program the ability to shift from one configuration to another, based on the values of the monitored parameters, are MONTOR and SHIFT. Figure 4.14 indicates the flow diagram for subroutine MONTOR. The subroutine that determines the amount by which the time step should be reduced, after a change in the configuration has been indicated, is DTPRIM. The flow diagram for this subroutine is seen in Figure 4.16.

(a) Subroutine MONTOR

Subroutine MONTOR checks the forces and/or the displacement differences at the three clearances to determine if any has become negative. As indicated by the shifting flow diagram, Figure 2.6, only three quantities must be monitored at each clearance, either forces and/or displacements differences. When at least one of the values becomes negative, subroutine DTPRIM is called.

(b) Subroutine DTPRIM

Subroutine DTPRIM linearly interpolates for a best estimate of the time the monitored value became negative, i.e. when the system changed to a different configuration. The calculated time

is stored in array DTP. If more than one change to the system configuration occurs in a single time step, the first change is chosen, i.e. the minimum value among the elements of the array DTP. The clearance index number that causes the system to change is then sent to subroutine SHIFT.

(c) Subroutine SHIFT

Subroutine SHIFT, described in section 3.D returns the new configuration based on the parameters sent by MONTOR.

6. Subroutine INPUT

This subroutine reads as data the formal parameters that are frequently varied. Figure 4.17 presents the flow diagram for this subroutine. These parameters include revolutions per minute (RPM) of the cam, clearance indicators (CLEAR, PIN 2&8) printing indicators (MPRINT, WTDEG, KPRINT, NCOUNT, IPRINT), output data indicators (IDRAW, AGTOUT, CARDS, TAPE IPLOT) and the tipe step (DELT). Graph titles are read into the arrays XNAME, YNAME, LABEL, and TITLE. The array TITLE is modified to indicate programs without a valve seat. The array LABEL is modified to indicate a two clearance problem. The lengths of the member elements are assigned (REALTH, AGTLTH) and the lever arm of the rocker (ROCKER) is calculated.

7. Subroutine CONMAT

This subroutine establishes two of the program generalization arrays, CONFIG and IDP. Figure 4.18 shows the flow diagram. The routine is called only once during the program execution. The coding for both arrays may be easily expanded to provide for problems with more than three clearances.

8. Subroutine INITAL

This subroutine initiates the starting values of displacement, velocity, acceleration, and force for the system coordinates. The starting configuration VII, requires that layer seven matrices be used for all calculations. Subroutine DINVS is called to establish the inverse stiffness matrix necessary for the displacement calculations. The program variables for the follower are set equal to the cam values and those for the left rocker arm are set equal to the pushrod values by calling subroutine ADJUST.

9. Subroutine CAM

This short subroutine calculates the displacement, velocity, and acceleration at the cam (coordinate 9) for the given ωt. The flow diagram is presented as Figure 4.15. The displacement is given by the equation:

$$q_9 = .5 \sin \omega t + \text{XZERO}$$

This routine is called at least once, every time increment from subroutine THESIS. Additionally cam is called by subroutine BKSTEP for intermediate values of ωt.

10. Subroutine BRANCH

This subroutine is called prior to making calculations for a new time step. Figure 4.19 indicates the flow diagram. The eight sections of the routine initialize the system parameters corresponding to its present configuration. Additionally the dimension parameter ISYS is set from the array IDIM. A correction ot the above established acceleration is accomplished when the array IMPACT indicates that a

clearance displacement difference is zero. The adjustment of the acceleration produces a corresponding increase in the later calculated force.

11. Subroutine ADJUST

This short subroutine equivalences the displacement, velocity, and acceleration quantities at a system coordinate to those at another system coordinate. For example, when contact exists between cam and follower the above named quantities for coordinate one are set equal to those for coordinate nine. Figure 4.20 presents the flow diagram.

12. Subroutine FIXFOR

This subroutine is primarily used to zero the calculated forces on the basis of the system configuration. This is necessary to eliminate small negative or small positive forces from being plotted. This routine has no effect on the program calculations. Its flow diagram is seen as Figure 4.21.

13. Output Subroutines

(a) Subroutine AGT

The subroutine AGT outputs data on punched cards or magnetic tape for use in the SDS-9300 program calculations of the graphic display. The flow diagram is indicated in Figure 4.22. This subroutine first adjusts the system displacement coordinates by an amount XZERO so as to present the initial position of the graphics display as horizontal. In addition to the displacement coordinates, reference values are sent for the cam rotation and the valve seat position.

(b) Subroutines ARRAY, GRAPH, GPRINT, OLPLOT

The subroutine ARRAY is called for the first value of each whole degree and establishes the array FPLOT, which is used for the plotting of graphs on the CALCOMP or offline printer. Figure 4.23 presents the flow diagram. Subroutine GRAPH is the last subroutine called by THESIS after the desired number of degrees has been evaluated. This subroutine, as seen in the flow diagram, Figure 4.24, organizes the calls to the offline printer subroutine OLPLOT for IPLOT equal to YES and/or the CALCOMP subroutine GPRINT for IDRAW equal to YES. The flow diagram for subroutine GPRINT is indicated in Figure 4.25. Subroutine OLPLOT is a modified version of the Computer Center provided routine UTPLOT. One of the modifications provides for an expansion of the graph output to a maximum of six pages. The number of pages desired is assigned to the program parameter, IP, by the user. Additionally the users must adjust the dimensions of the main program array, GRID to conform with the desired pages. GRID must be dimensioned as LINES by 81 where the dimension parameter LINES is:

$$LINES = (80)(IP) + 1$$

(c) Subroutine DRU--

The many subroutines beginning with DRU-- were written for plotting graphs on the CALCOMP plotter by Winfrey. This package also contains computer center provided routines, indicated by an asterick in Figure 4.9.

(d) Subroutine LIST (DEBUG)

Subroutine LIST outputs the numerical formal parameters on the offline printer for either "debugging" purposes or as reference values when viewing the graphs/graphics.

(e) Subroutine PRNTMX

Subroutine PRNTMX outputs all the system matrices, when the program parameter MPRINT is equal to YES.

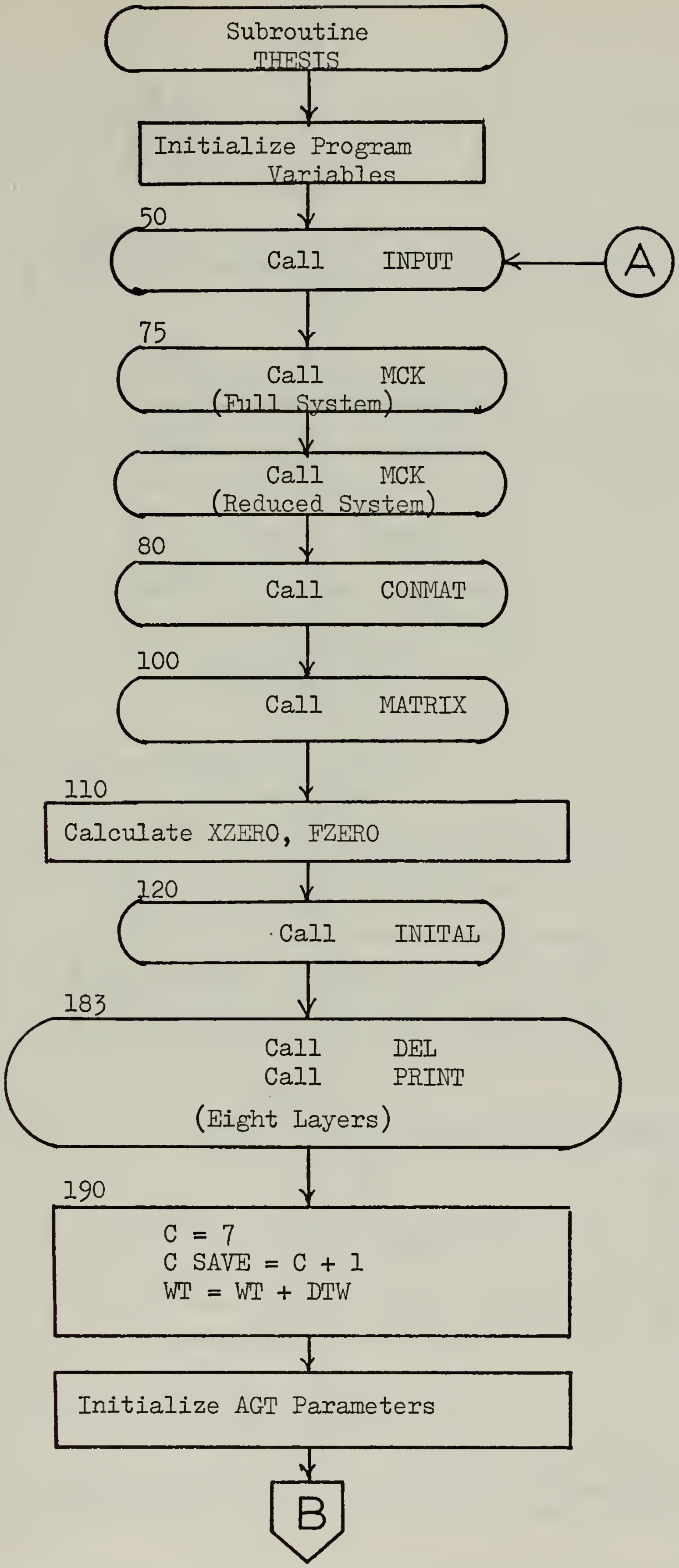


Figure 4.10 Subroutine THESIS Flow Diagram

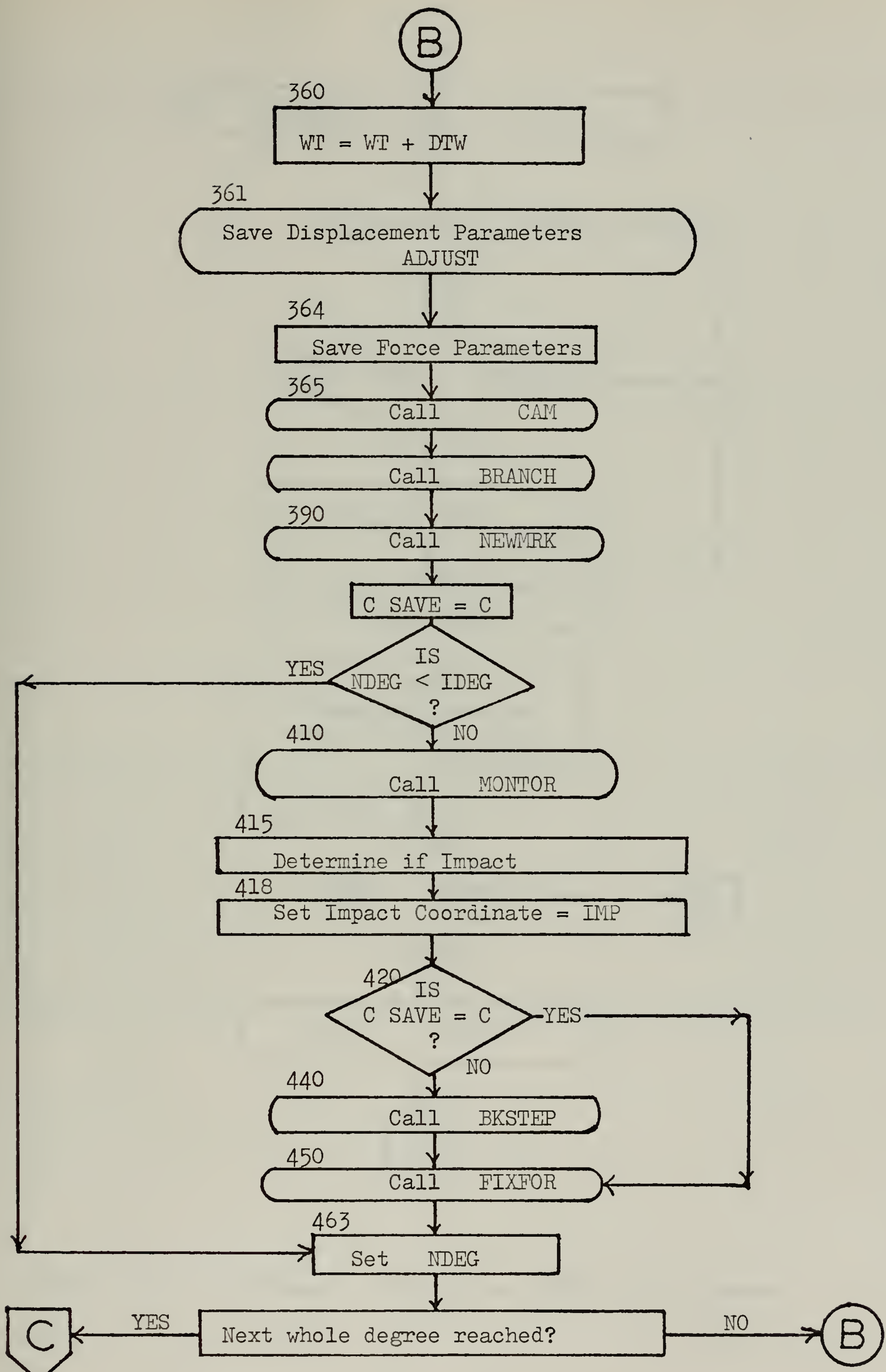


Figure 4.10 Continued - Subroutine THESIS Flow Diagram

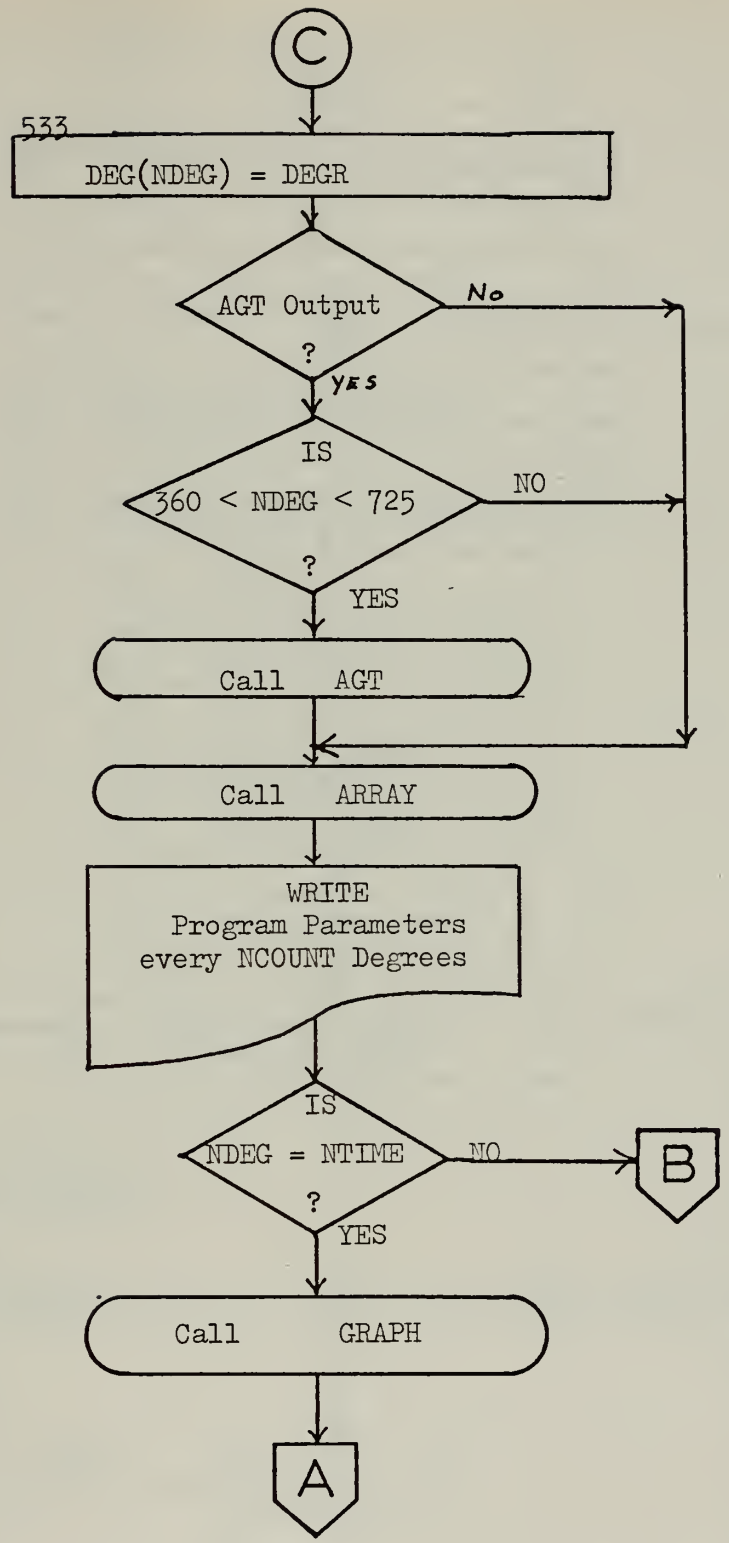


Figure 4.10 Continued - Subroutine THESIS Flow Diagram

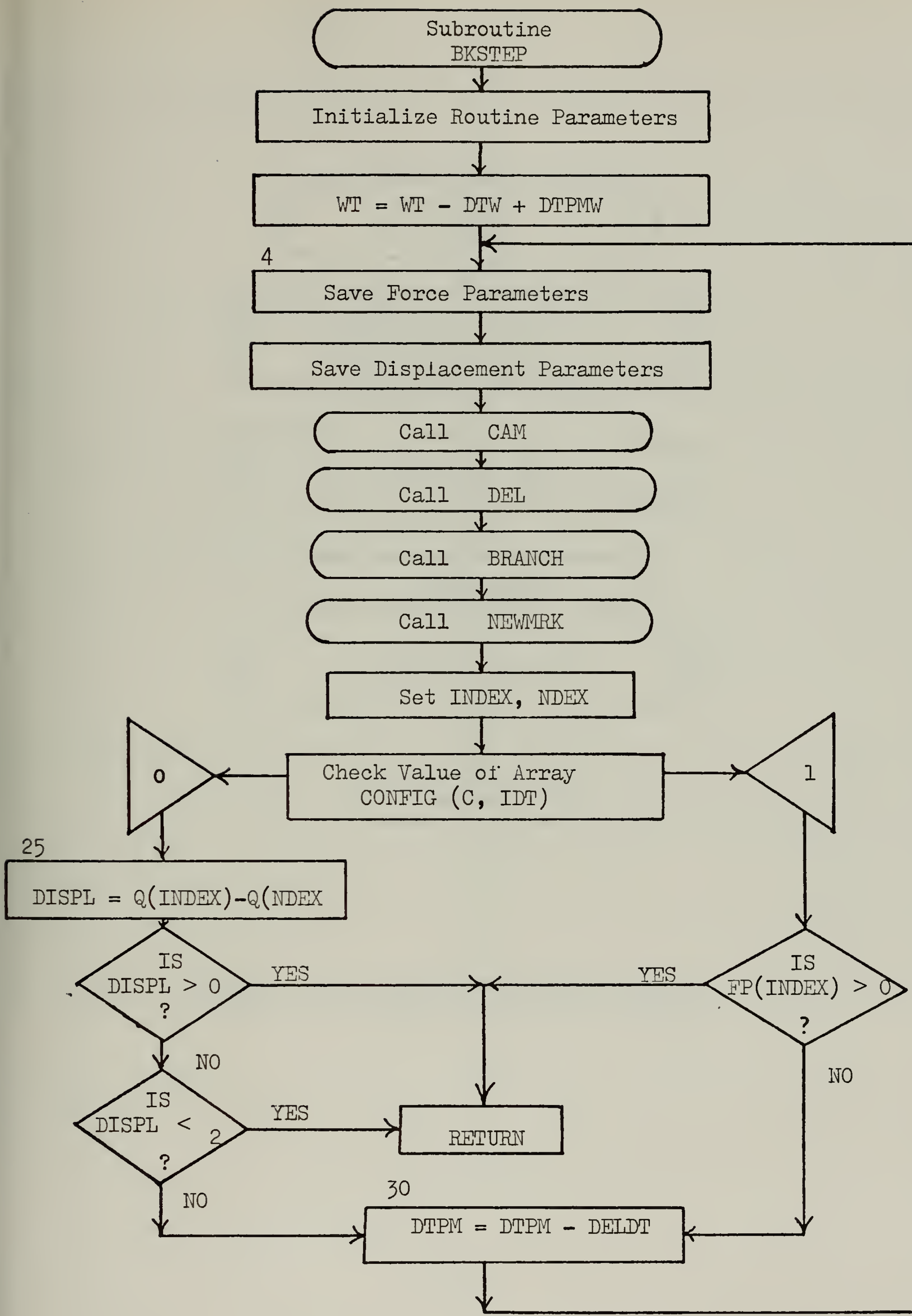


Figure 4.11 Subroutine BKSTEP Flow Diagram

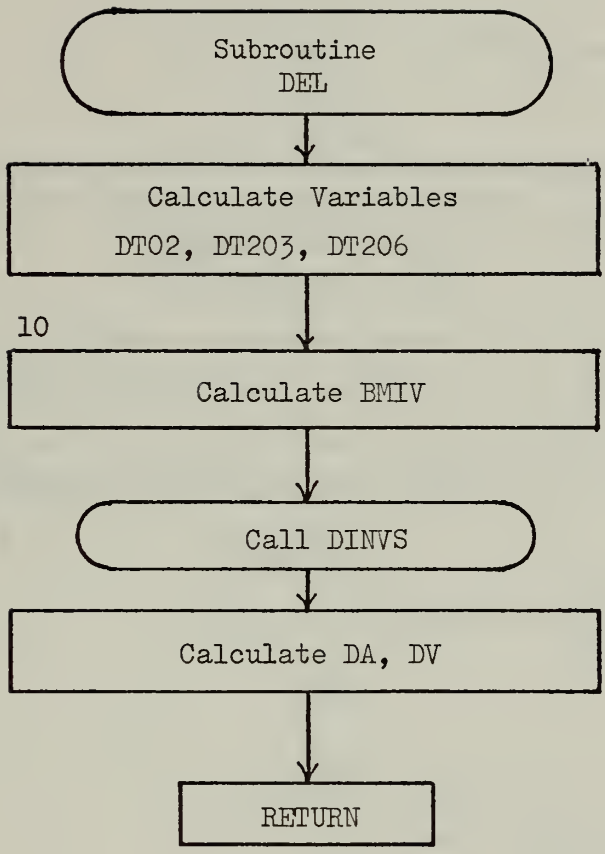


Figure 4.12 Subroutine DEL Flow Diagram

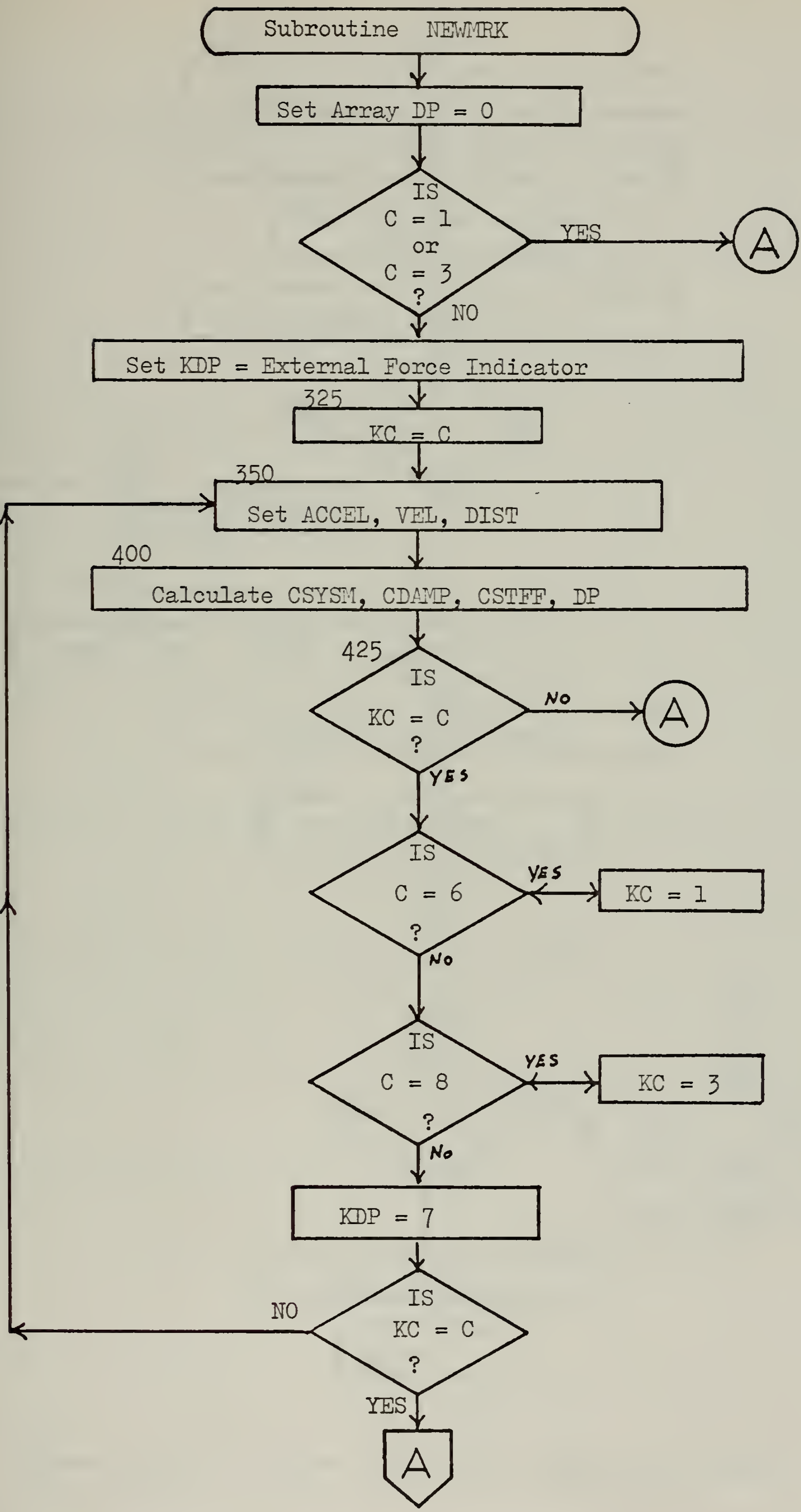


Figure 4.13-1 Subroutine NEWMRK Flow Diagram

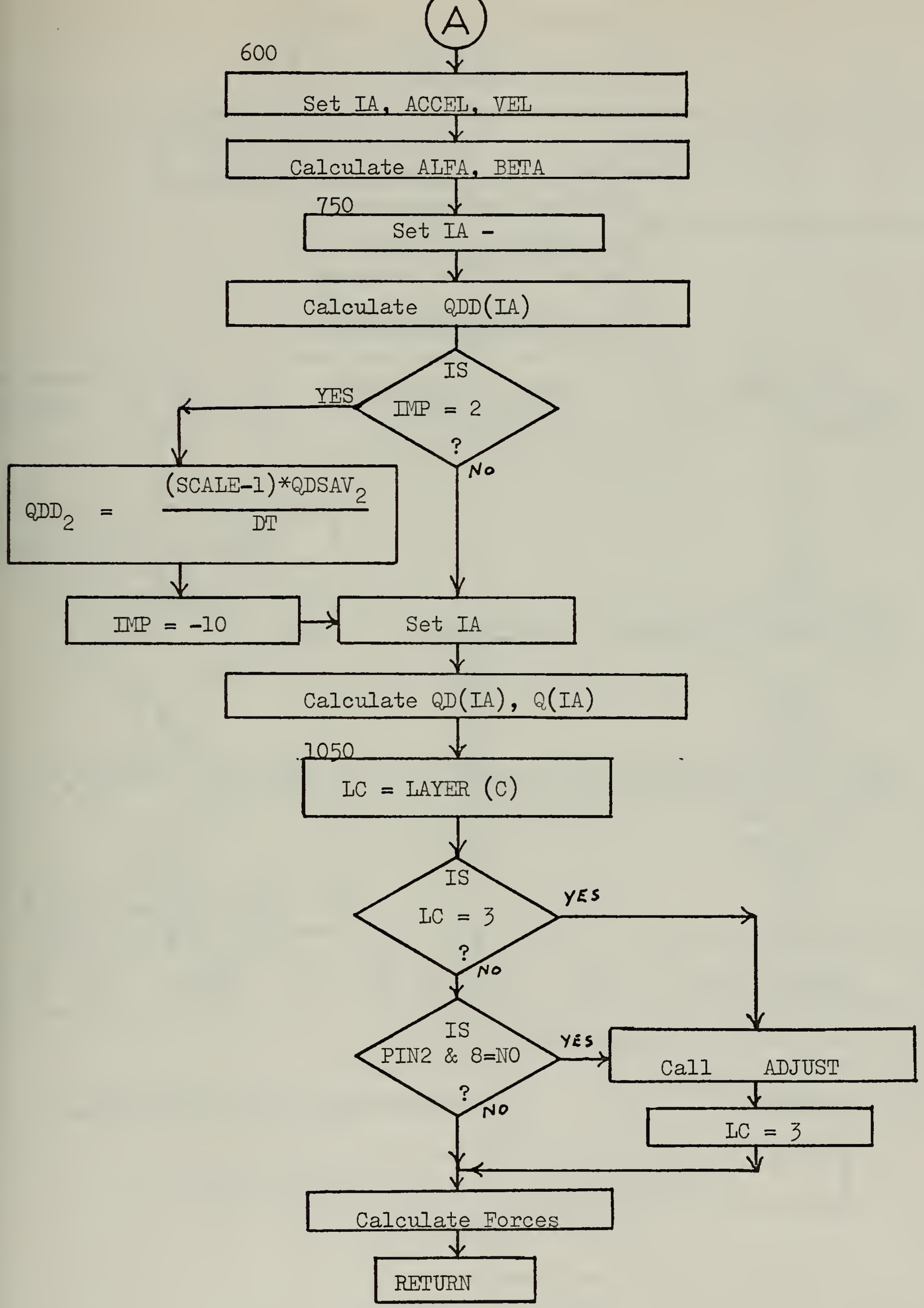


Figure 4.13-2 Continued - Subroutine NEWMRK Flow Diagram

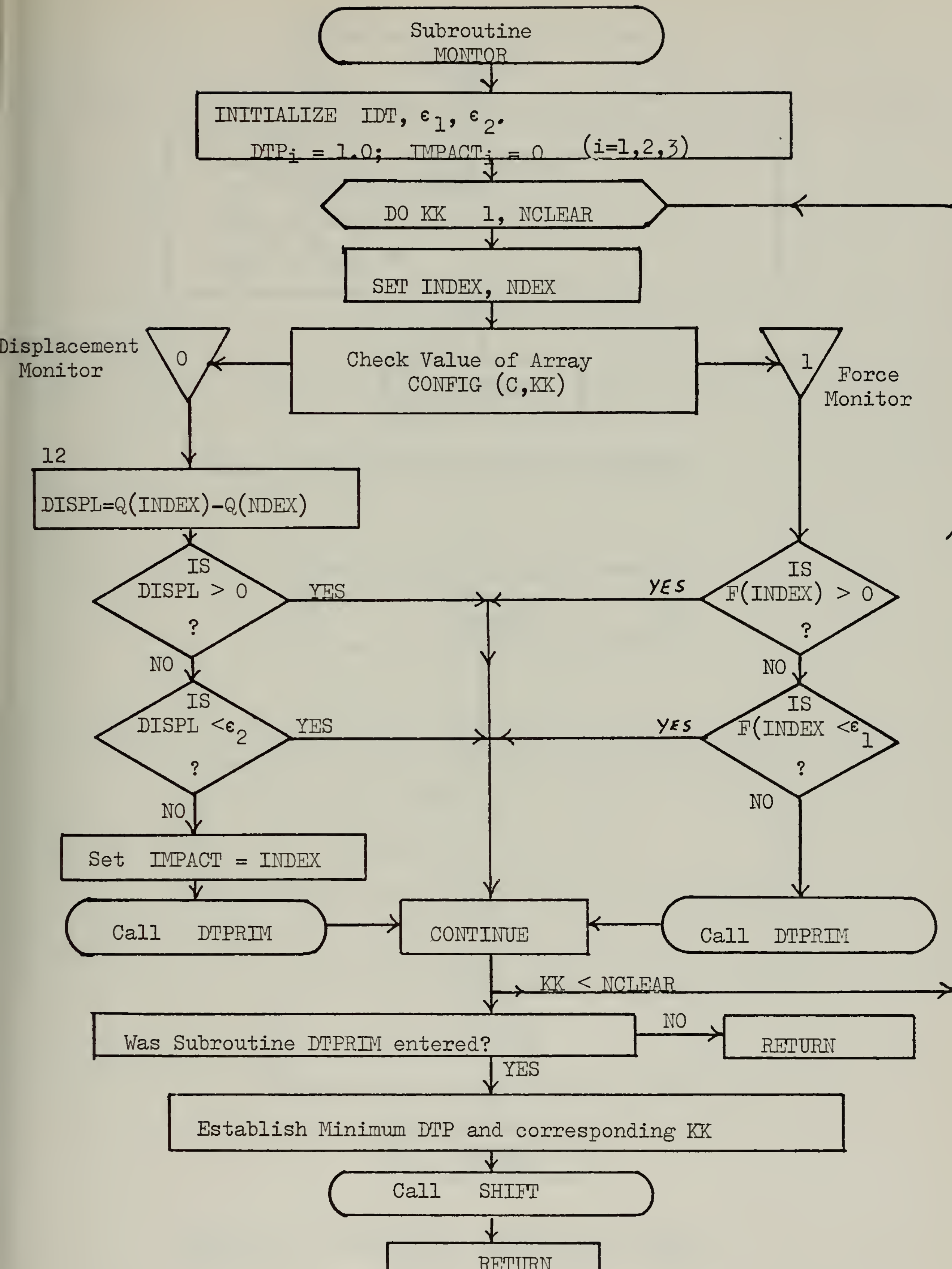


Figure 4.14 Subroutine MONTOR Flow Diagram

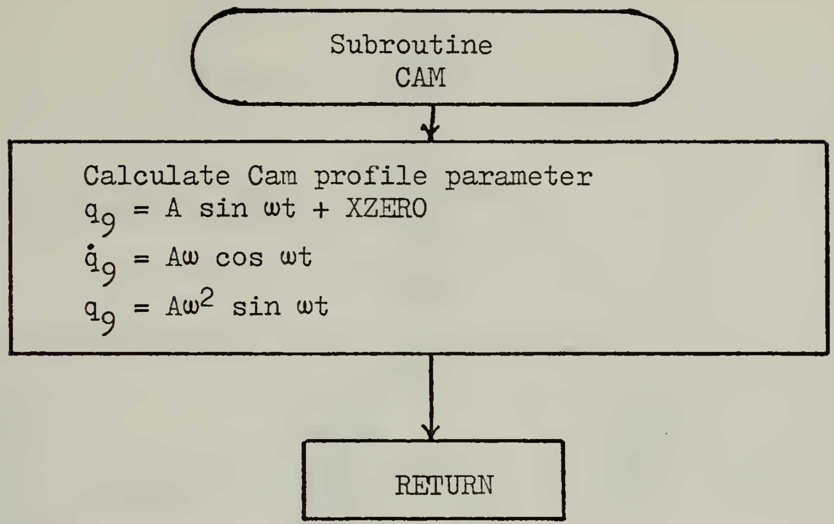


Figure 4.15 Subroutine CAM Flow Diagram

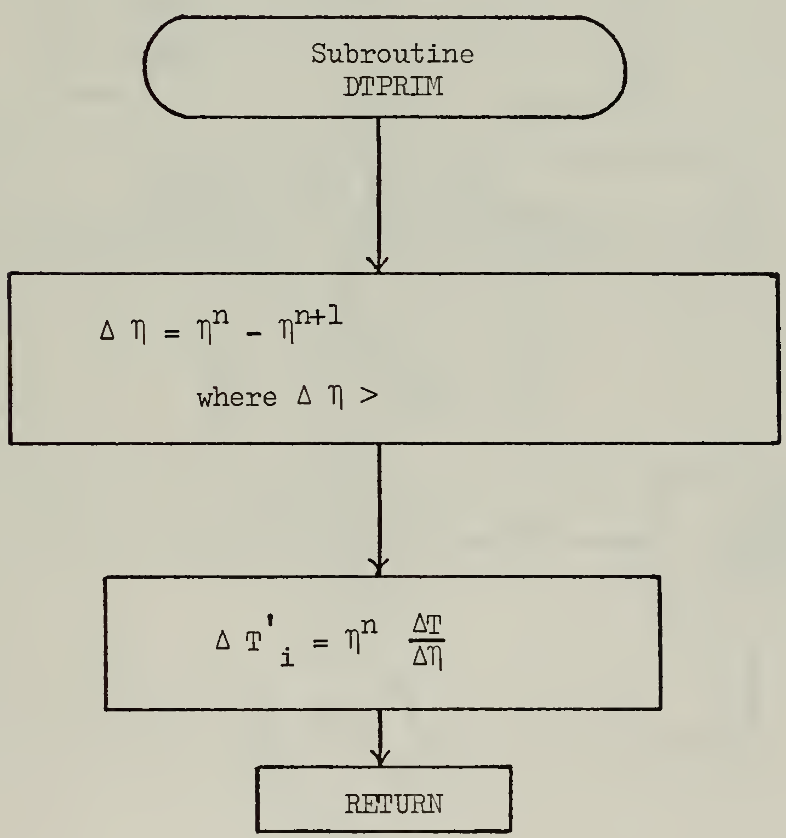


Figure 4.16 Subroutine DTPRIM Flow Diagram

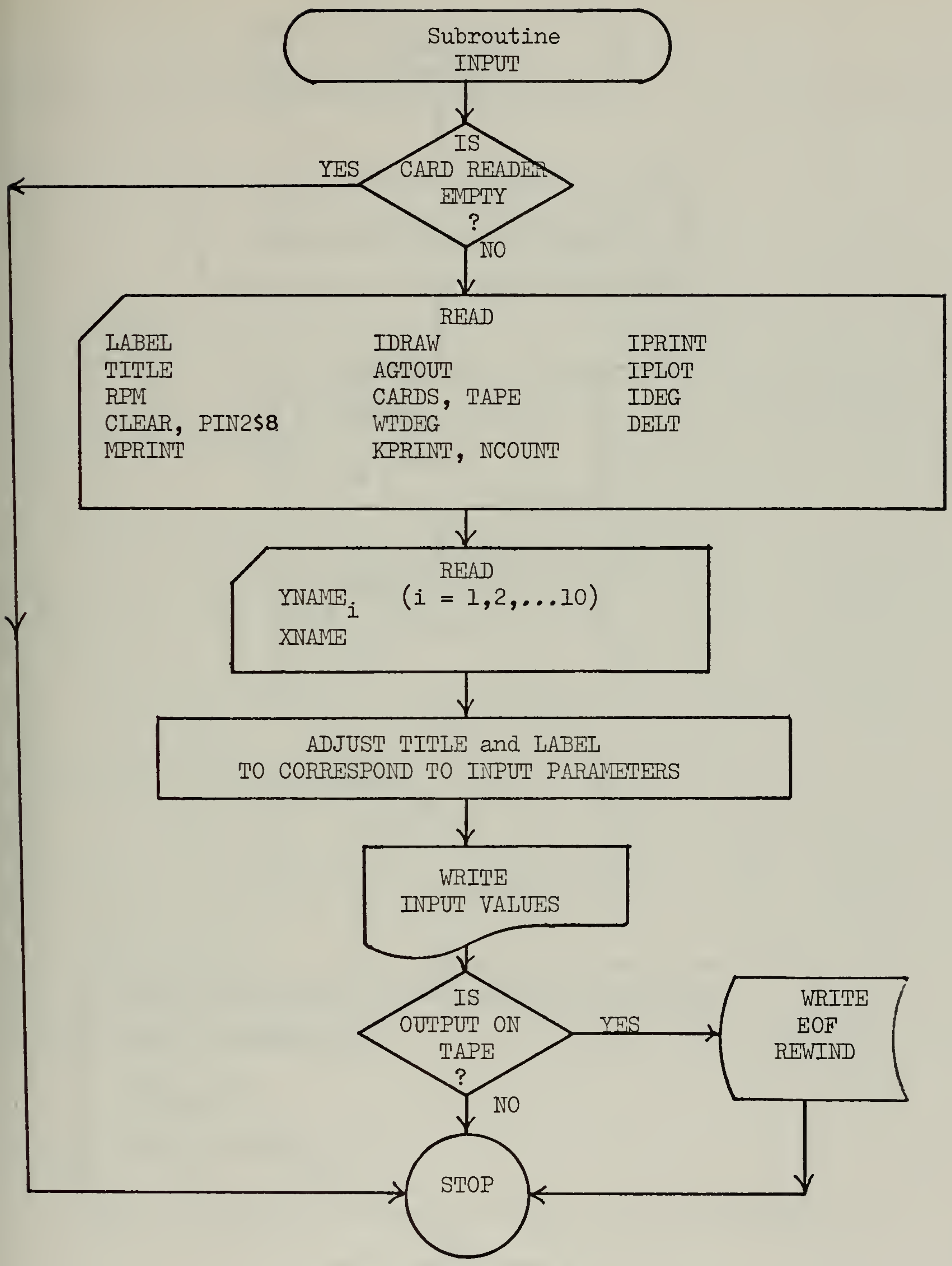


Figure 4.17 Subroutine INPUT Flow Diagram

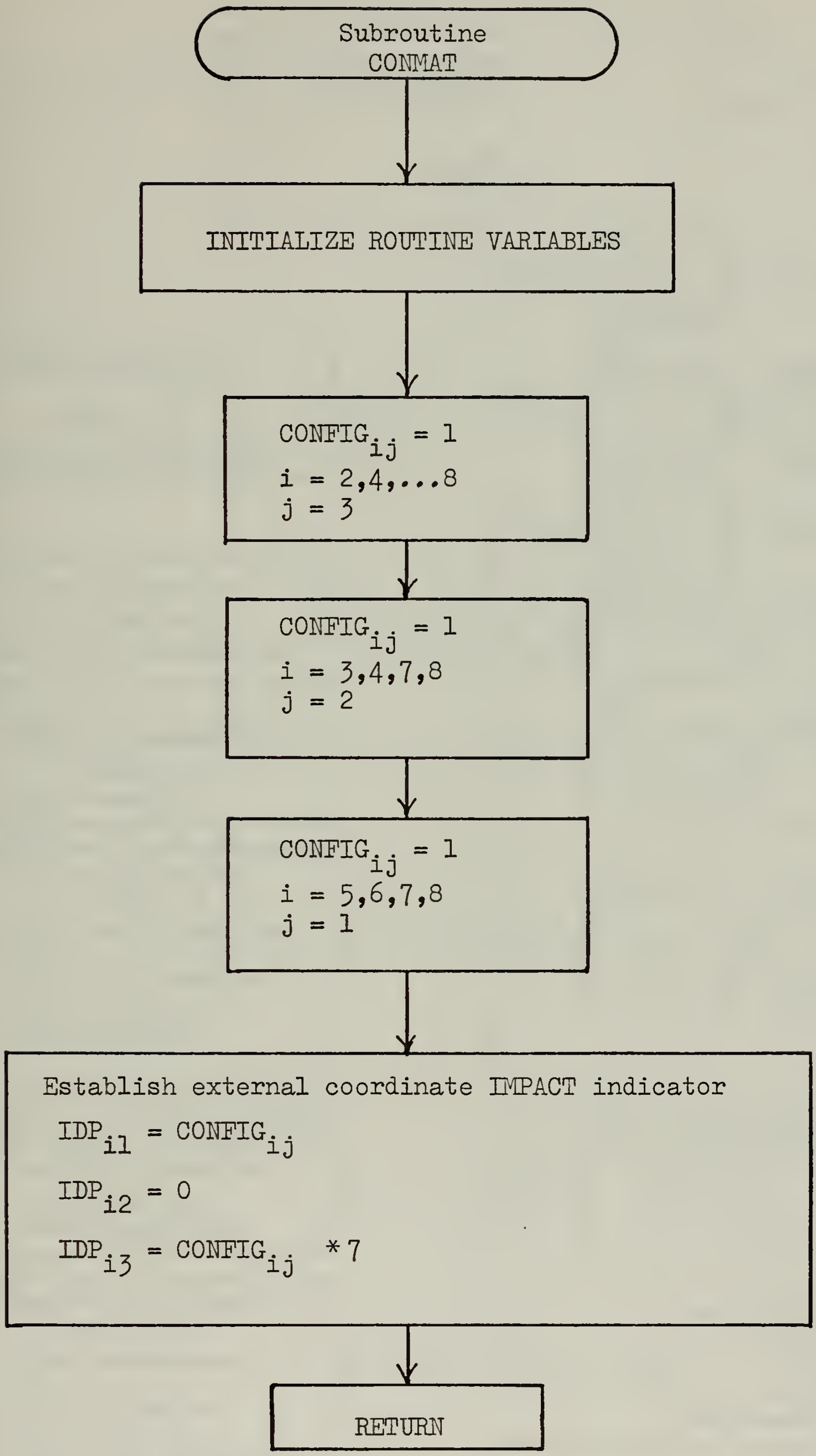


Figure 4.18 Subroutine CONMAT Flow Diagram

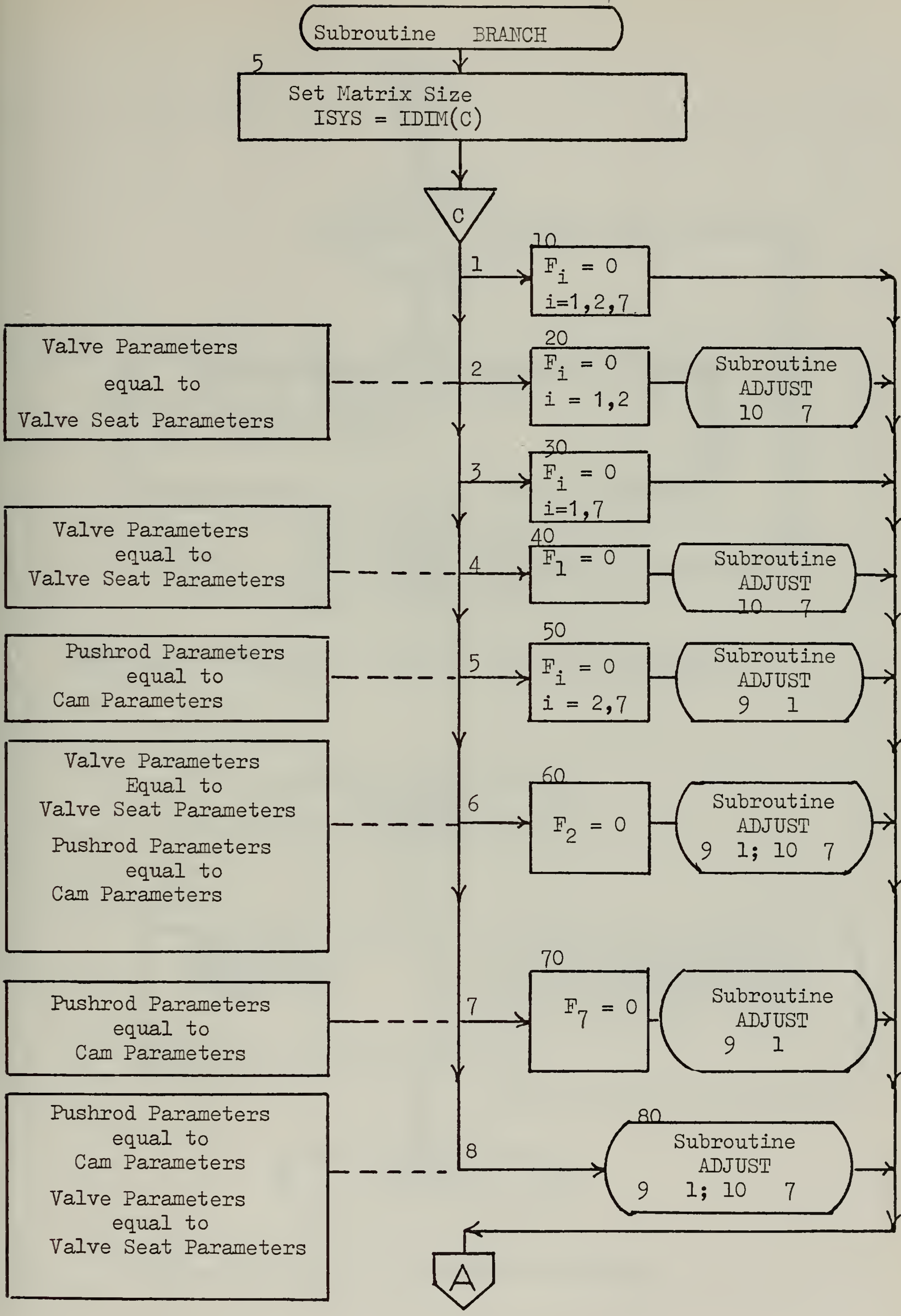


Figure 4.19 Subroutine BRANCH Flow Diagram

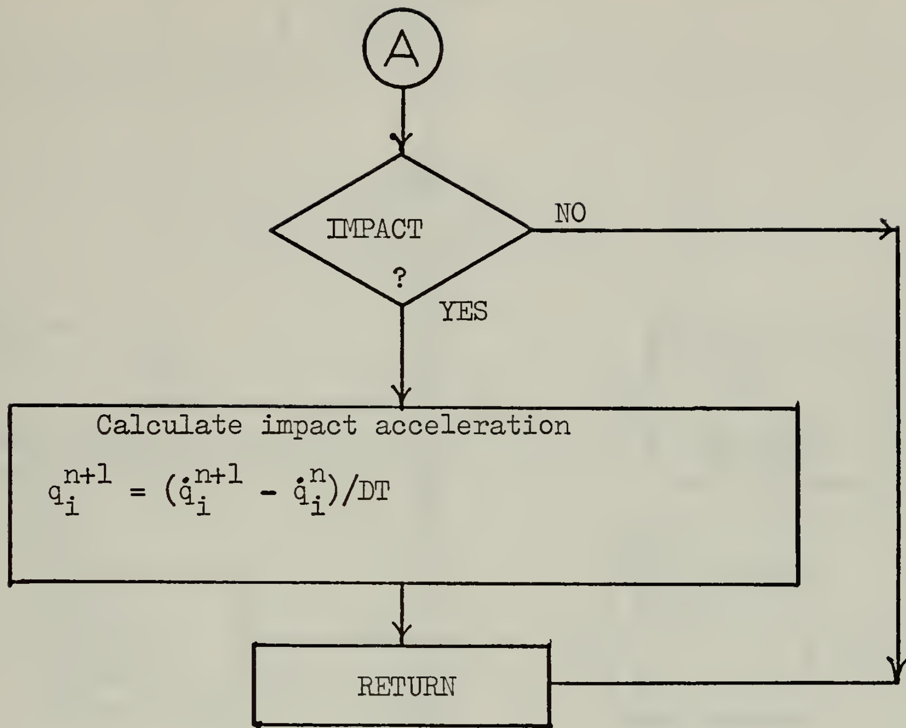


Figure 4.19 Continued - Subroutine BRANCH Flow Diagram

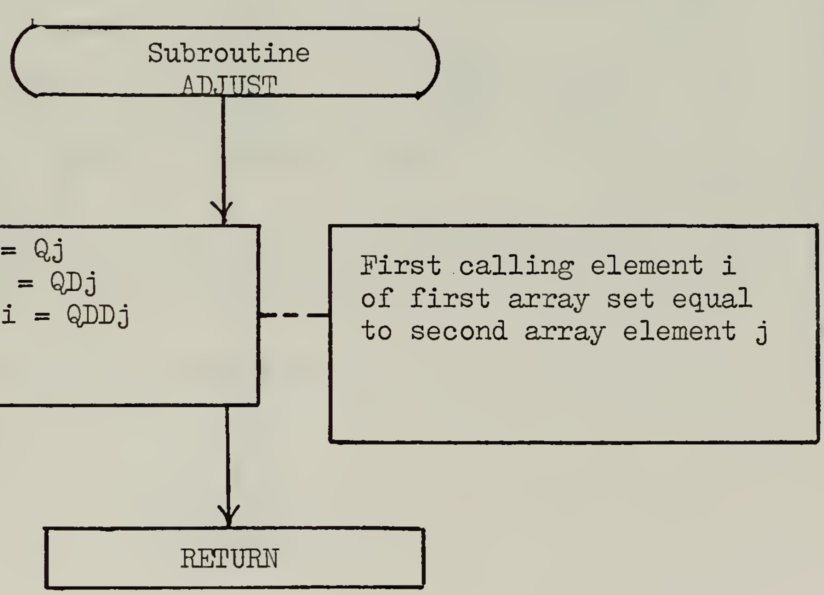


Figure 4.20 Subroutine ADJUST Flow Diagram

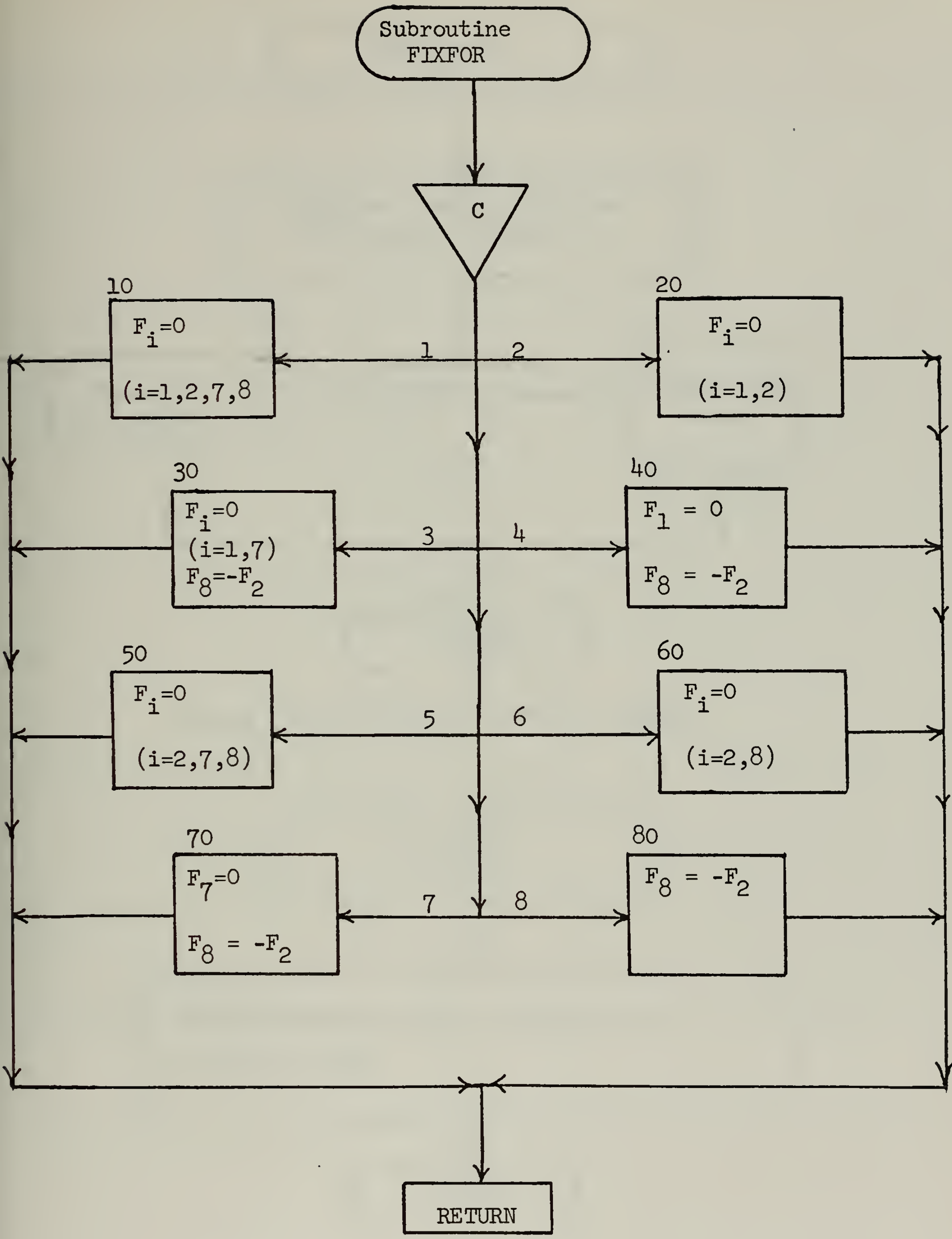


Figure 4.21 Subroutine FIXFOR Flow Diagram

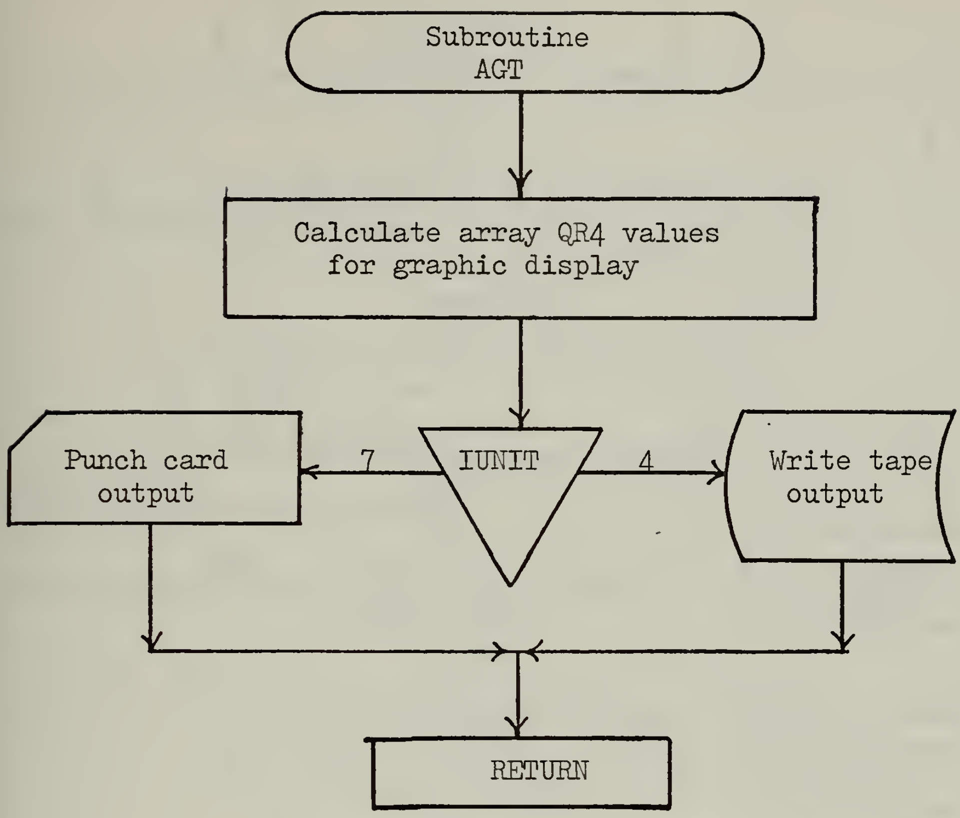


Figure 4.22 Subroutine AGT Flow Diagram

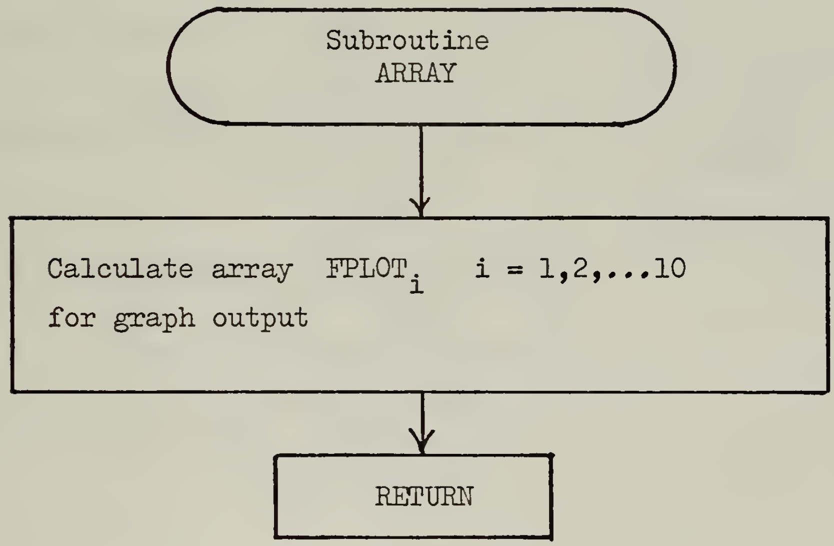


Figure 4.23 Subroutine ARRAY Flow Diagram

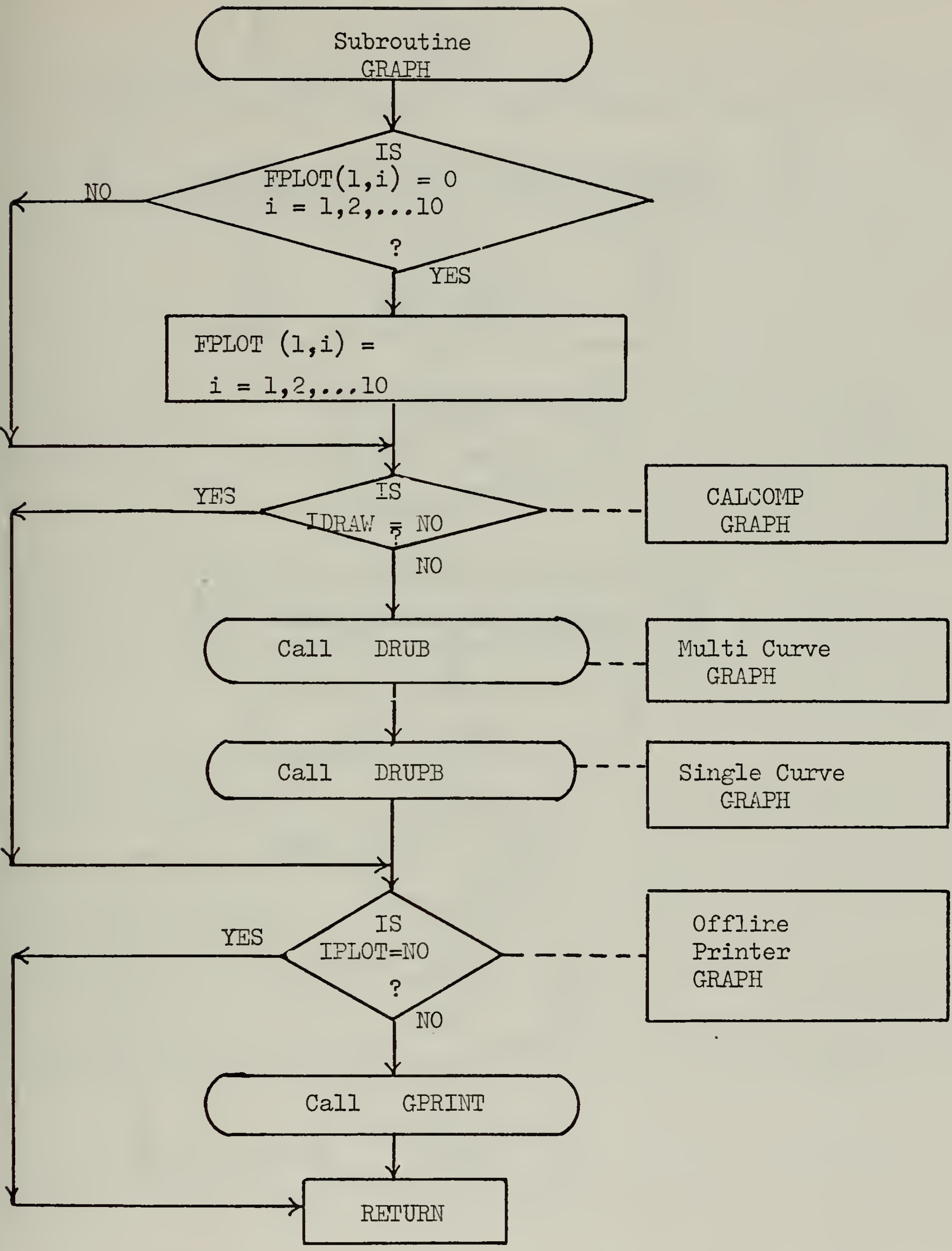


Figure 4.24 Subroutine GRAPH Flow Diagram

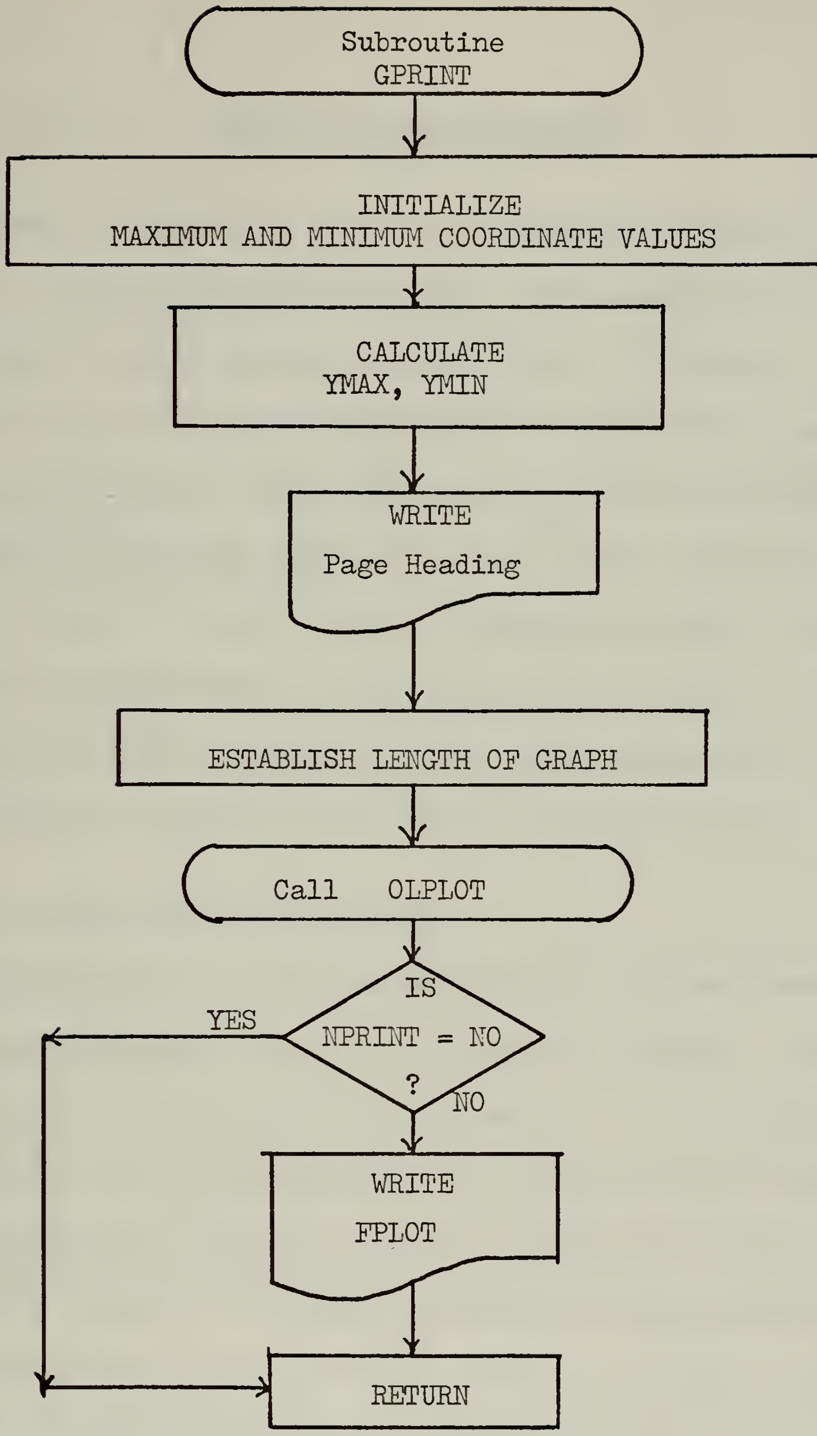


Figure 4.25 Subroutine GPRINT Flow Diagram

V. SDS-9300 COMPUTER PROGRAM

This computer program is coded in SDS FORTRAN IV, (Ref. 5-7), to run on a Scientific Data System Model 9300 computer interfaced with an ADAGE Graphics Display Terminal, Model 10 (AGT-10). This program provides the user the capability to observe the dynamic response of 360 degrees of cam rotation of the valve train assembly. The displacement data generated by the previously described program is the major input. Finite element techniques are used to reformulate the input displacement data to provide for the pictoral responses of bending and axial deformations. Due to the limited core storage of this computer, overlaying techniques have been employed.

A. METHOD OF ELASTICITY CALCULATIONS

As shown in Figure 2.4,the system consists of five elements. Three of these elements, (1,4, and 5) are axial members. The remaining two, (2 and 3) are bending members. Thus the elastic mode shapes for each of the elements as a function of the distance along the member will yield the required deformations. R. C. Winfrey, reference (1) Chapter V, has done considerable work in the formulation of the required mode shapes by finite element techniques. The mode shapes for the axial elements as a function of x are:

$$\Phi_1 (x) = \frac{x}{L} \tag{5.1}$$

$$\Phi_2 (x) = \frac{x}{L} - 1 \tag{5.2}$$

where L is the longitudinal dimension of the bar.

The mode shapes for the bending elements as a function of x are:

$$\Phi_1(x) = \frac{1}{L^3}(L^3 - 3x^2L + 2x^3) \tag{5.3}$$

$$\Phi_2(x) = \frac{1}{L^3}(3x^2L - 2x^3) \tag{5.4}$$

$$\Phi_3(x) = \frac{1}{L^2}(xL^2 - 2x^2 L + x^3) \tag{5.4}$$

$$\Phi_4(x) = \frac{1}{L^2}(x^2L - x^3) \tag{5.6}$$

where L is the longitudinal dimension of the bar.

The system displacements are calculated by the IBM-360 program and sent as data to the SDS-9300. These displacements and the above mode shapes are superimposed to yield the required rigid body motion and elastic deformations.

B. PROGRAM STRUCTURE

1. General Description

The main program controls the program execution. The calling order is presented in Figure 5.1. The subroutines indicated by an asterick are supplied by the computer laboratory, and are used to manipulate and initialize data sent to the ADAGE Graphics Terminal. The SDS computer total core available for program execution, exclusive of the control and real time monitoring functions, is approximately 18 K bytes. Since the total program length greatly exceeds the core available, a program overlay technique consisting of storing only active subroutines during execution is employed. This achieves maximum use of the main core storage and allows the program to execute.

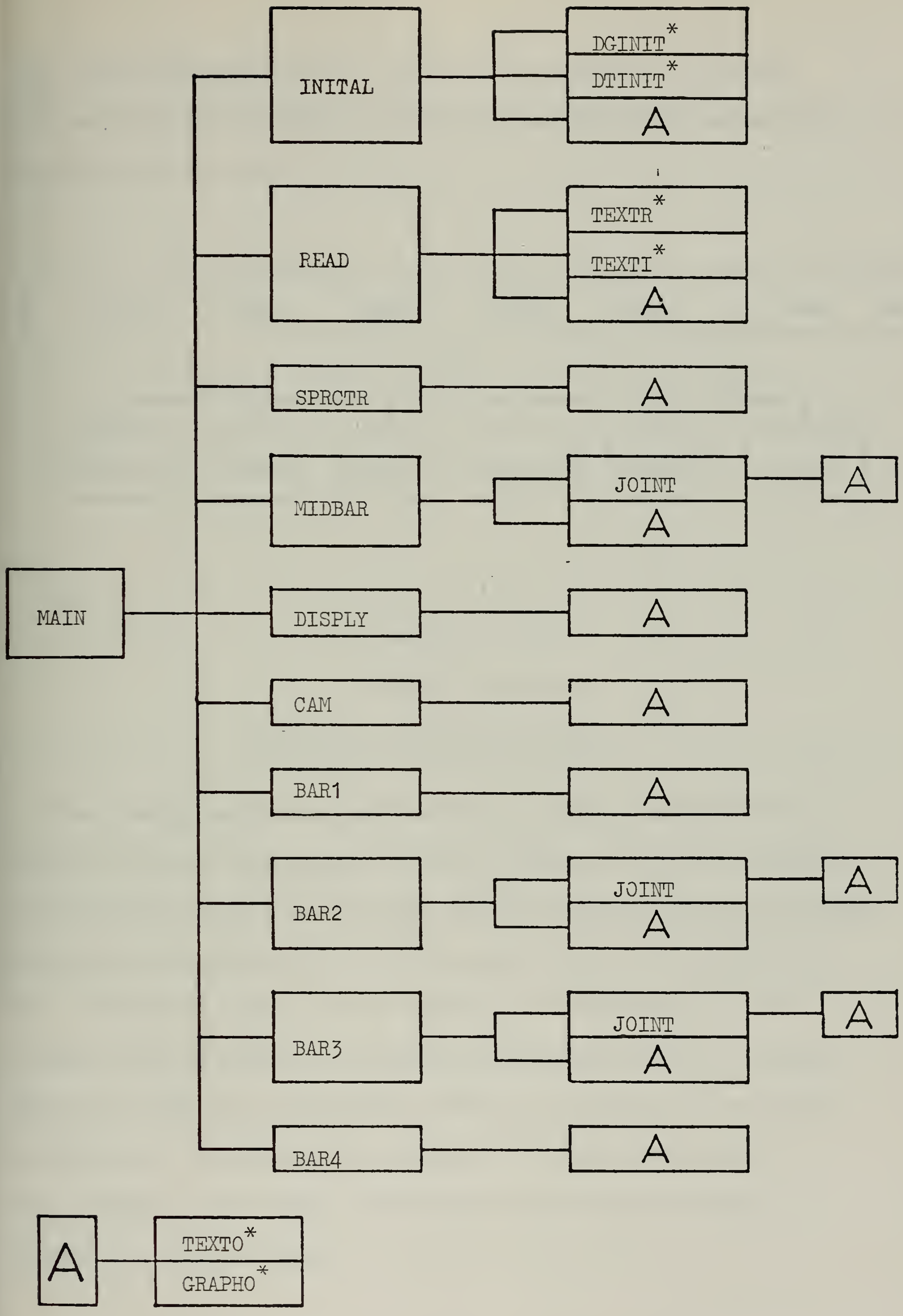
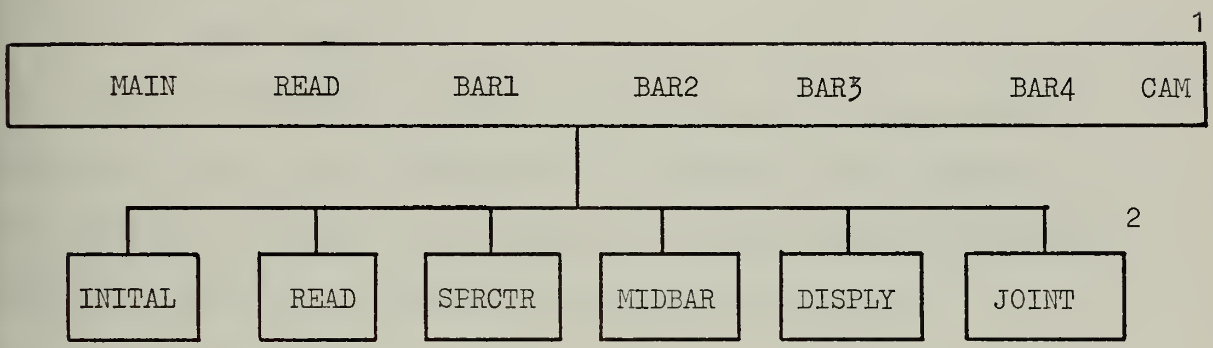


Figure 5.1 SDS-9300 Computer Program Structure

The overlay tree structure is depicted in Figure 5.2. Although necessary for this program, overlaying significately increases the program execution time.



1. Remains in core

2. Overlay structure

Figure 5.2 Overlay Structure

The bowing and thinning of the axial elements, as seen on the graphics display photographs, have been added as an aid in observing the deformations of these members. The mode shapes given by equations 5.1 and 5.2 are only valid for the elastic line of a one dimensional bar. However the small uniform extension and compression of the element were impossible to satisfactorily reproduce on the display. Thus the bowing and thinning were added as a function of the axial deformation. The thinner the bar, the more axial compression it has undergone. Conversely, the thicker the bar the more axial extension it has undergone.

2. <u>Sense Lights</u>

The SDS computer being a hands-on system, has several physical sense lights. When lit, logic within the program may alter computer instructions as desired.

a. Sense Light 1.

Activation of sense light one branches the execution to the head of the program. Execution then continues and variables are reinitialized as coded in the program. New data points may also be read in, displaying a different set of displacement data.

b. Sense Light 2.

During program execution this sense light indicates that the display will utilize ADAGE Graphics Terminal, number two. When not lit, the SDS computer is interfaced with the AGT unit number one.

c. Sense Light 4.

This sense light branches the program to a convenient point to reinitialize the originally coded program parameters. This is accomplished by activation of the SDS control console teletype from which the user may input any program variables that appear in the namelist statement.

d. Sense Light 6.

Activation of sense light sixcalls the subroutine TEXTR for each cam rotation increment. This enables the motion of the display to be halted for further study at any point of interest in cam rotation. Depression of the return key on the AGT terminal will advance the display to the next display of cam rotation.

C. MAIN PROGRAM

The main program initializes parameters, establishes the data arrays and contains the calculation loop. Prior to the execution of the program the desired sense lights should be activated. The program is halted almost immediately after execution by an INPUT statement. This statement allows the user to have positive control over assignment of the AGT unit. The designation of the wrong unit at this point will case a core image failure. The program then must be reloaded. Therefore at this point the 9300 teletype control console should be checked to insure the correct unit is listed in the message to the user. To proceed with the program execution a "*" must be typed on the control console followed by the carriage return key.

Subroutines are called to display the fixed elements of the model. The two arrays that reproduce the circular cam are pre-calculated. Next the frequency ratio is calculated and displayed. The cycle loop is then entered where subroutines are called to display the movable elements. This loop is incremented by a specified amount of cam rotation until 360 degrees of rotation has been reached. At this point the cycle is repeated. For each cam increment, the sense lights are tested.

D. SUBROUTINES

The function of each of the subroutines and a description of their execution are included. Each subroutine is listed in its entirety in the SDS-9300 Computer Program Listing.

1. <u>Subroutine INITAL</u>

This subroutine initializes the text and graphics directory arrays, DTINIT and DGINIT, respectively. The text response, TEXTR is initialized to block number one at display location 39, 46. The title display is encoded and presented on the AGT. The graphic data blocks are initialized to zero, assigned the appropriate first word header (IHEAD), and then assigned sequence numbers in DGINIT for later use.

2. <u>Subroutine READ</u>

This subroutine reads the title and label for the program and prints them on the offline printer. The program parameters are read, (XZERO, AMP, RPM, WN, ISEAT, and PIN28) and also printed for reference. The user is then questioned from the AGT display, Photo 5.1.

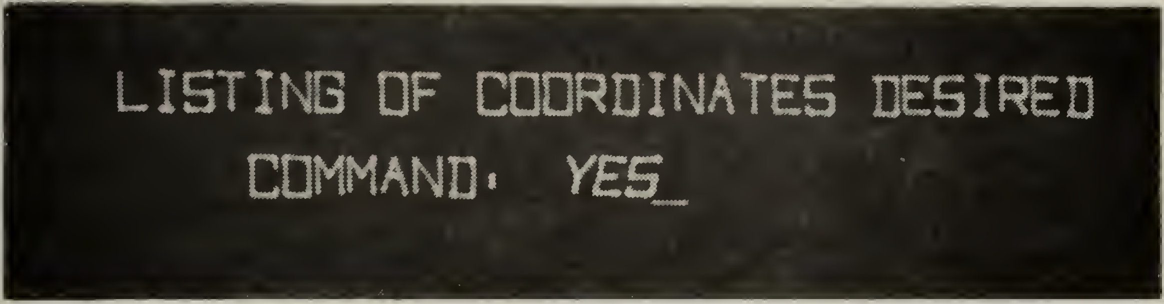


Photo 5.1 AGT QUESTION

An affirmative response will list the IBM-360 displacement data on the offline printer. The response is entered via the graphics terminal.

The displacement data is next read from magnetic tape or punched cards, depending on the control specification at the time the program was loaded. The array parameters read are, Q1 through Q9, OMEGA, and CONFIG. The Q's are as defined in the IBM program but

modified by the amount XZERO. OMEGA is the amount of revolution of the cam in radians. CONFIG is the corresponding configuration number. Next the displacements are printed, if desired. Finally the title screen is nulled and control is returned to the main program.

3. Subroutine SPRCTR

This subroutine is only called once for a given problem. The display for the rigid portion of the spring is calculated and displayed. The valve seat position and geometry is calculated. The parameter ISEAT is then checked to determine whether the problem considers contact at the valve seat. For those problems with no contact, the valve seat location is adjusted to better picture the position. Finally the valve seat is displayed and control returns to the main program.

4. Subroutine MIDBAR

This subroutine is called once. Subroutine MIDBAR calculates and displays the fixed pivot point.

5. Subroutine DISPLY

This subroutine is called once during each cam increment. The function of this routine is to display the text for the system configurations.

6. Subroutine CAM

Subroutine CAM is called once during each cam increment. This routine calculates the cam position, given the cam rotation (OMEGA) and displacement of the top of the cam (q_9).

7. Subroutine BAR1

This subroutine is called once during each cam increment. Coordinates q_1 and q_8 are used to calculate the position of the pushrod. Equations 5.1 and 5.2 calculate the deformations. The pushrod is then displayed on the AGT.

8. Subroutine BAR2

This subroutine is called once during each cam increment. Coordinates q_2, q_3, and q_6, and equations 5.3, 5.4, and 5.5 are used in the calculations of position, geometry, and deformations of the left rocker arm. The parameter PIN28 is checked to determine whether the pin should be displayed at clearance two. The left rocker arm and pin, if required, are displayed on the AGT.

9. Subroutine BAR3

This subroutine is called once during each cam increment. Coordinates q_4, q_5, and q_6, and equations 5.3, 5.4, and 5.5 are used in the calculation of position, geometry, and deformations of the right rocker arm. Calculation of the position and image of the rocker arm- valve stem- valve spring joint pin is accomplished by subroutine JOINT. The right rocker arm and pin are then displayed on the AGT.

10. Subroutine BAR4

This subroutine is called once during each cam increment. Coordinates q_4 and q_7 are used to calculate the position of the valve stem. Equations 5.1 and 5.2 calculate the deformations. The valve-valve stem is then displayed on the AGT.

11. Subroutine JOINT

This subroutine is called once for the right rocker arm pin and, if required, once for the left rocker arm pin, during each cam increment. The pins' position and image are calculated and displayed on the AGT.

VI. NUMERICAL RESULTS

The precalculated system matrices for the physical properties assumed in chapter 2, are presented in this chapter. Numerical examples have been chosen for their interest and for comparison with one another. The relationship between examples is indicated in Figure 6.26. Example results are presented both as CALCOMP graphs and/or AGT display photos. Photo 6.1 is the title page from the graphics terminal display.

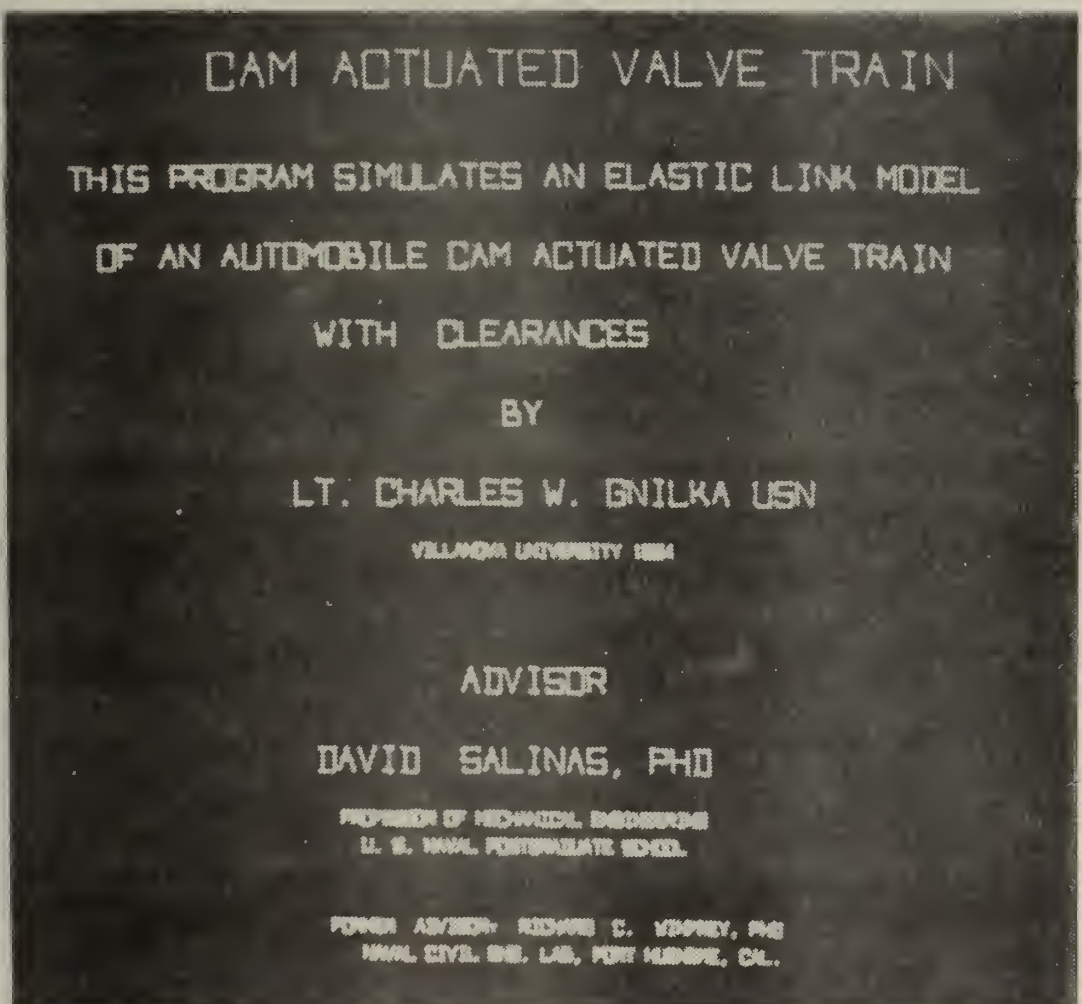


Photo 6.1 AGT Title Page

A. MODE SHAPES AND NATURAL FREQUENCIES

The mode shapes and natural frequencies presented in Figure 6.1 result from the eigenvalue problem for the full eight coordinate system. Figure 6.2 shows the values resulting from the seven coordinate reduced system, for contact at the rocker.

NATURAL FREQUENCIES ω_n (EIGENVALUES) x 10^3

0.0	63.6	638.7	0.886	263.8	1371.1	149.8	77.86

MODE SHAPES (EIGENVECTORS) IN COLUMNS [Φ]

46.3	0.0	0.0	0.0	0.0	0.0	0.0	-80.2
0.0	115.0	-109.5	-37.1	-57.2	-172.7	33.3	0.0
0.0	-131.3	581.0	37.1	149.1	1538.0	-56.0	0.0
0.0	21.5	-34.2	55.6	60.9	18.7	42.5	0.0
0.0	58.7	-499.4	37.0	347.7	376.9	-29.8	0.0
0.0	-70.9	-340.0	37.1	-146.7	381.4	24.6	0.0
0.0	-35.0	-18.9	-55.6	52.8	9.6	120.0	0.0
46.3	0.0	0.0	0.0	0.0	0.0	0.0	80.2

Figure 6.1 FULL SYSTEM MODE SHAPES AND NATURAL FREQ.

NATURAL FREQUENCIES ω_n (EIGENVALUES) x 10^3

0.691	73.6	521.9	25.9	247.1	1070.2	146.7

MODE SHAPES (EIGENVECTORS) IN COLUMNS [Φ]

28.96	-69.26	15.64	45.72	11.22	12.01	-8.39
28.95	64.03	-30.77	20.91	-20.84	-23.92	13.67
-28.95	-70.49	503.13	-30.45	136.63	799.47	-42.66
-43.37	9.09	-34.11	39.89	61.27	25.02	38.35
-28.90	32.45	-427.34	39.72	313.46	471.17	-45.62
-28.93	-42.34	-215.68	-1.53	-164.75	436.79	44.23
43.37	-18.08	-19.75	-42.97	57.06	12.96	118.61

Figure 6.2 REDUCED SYSTEM MODE SHAPES AND NATURAL FREQ.

B. PRE-CALCULATED SYSTEM MATRICES

The remaining system matrices are presented and grouped according to their layer (or configuration) number. The mass, stiffness, and damping matrices for layer one are unaltered and directly correspond to the full eight coordinate system eigenvalue problem. The mass, stiffness, and damping matrices in layer three are unaltered and directly correspond to the reduced seven coordinate system eigenvalue problem.

The pre-calculated matrices COLM, COLC, and COLK located in layers one and three are associated with the calculations for configurations six and eight respectively. Thus, these matrices are not listed in their correct program layer, but with the correct configuration with which they are used.

The order of presentation within each layer is: (1) Mass, (2) Damping, (3) Stiffness, (4) COLM, (5) COLC, (6) COLK, (7) BMIV, (8) DA, and (9) DV.

CONFIGURATION AND MATRIX LAYER = 1

MATRICES ARE 8X8

--- SYSM ---

1.55D-04	0.0	0.0	0.0	0.0	0.0	0.0	7.77D-05
0.0	7.70D-05	1.09D-05	0.0	0.0	-6.42D-06	0.0	0.0
0.0	1.09D-05	1.97D-06	0.0	0.0	-1.48D-06	0.0	0.0
0.0	0.0	0.0	2.01D-04	-2.44D-05	1.44D-05	-2.59D-05	0.0
0.0	0.0	0.0	-2.44D-05	6.66D-06	-5.00D-06	0.0	0.0
0.0	-6.42D-06	-1.48D-06	1.44D-05	-5.00D-06	8.64D-06	0.0	0.0
0.0	0.0	0.0	-2.59D-05	0.0	0.0	5.18D-05	0.0
7.77D-05	0.0	0.0	0.0	0.0	0.0	0.0	1.55D-04

--- DAMP ---

6.05D-01	0.0	0.0	0.0	0.0	0.0	0.0	-6.05D-01
0.0	1.32D 00	2.93D-01	5.54D-01	-1.69D-01	2.49D-01	-7.00D-02	0.0
0.0	2.93D-01	1.42D-01	1.00D-01	-3.70D-02	2.01D-02	-1.01D-02	0.0
0.0	5.54D-01	1.00D-01	1.51D 00	-3.36D-01	-5.39D-02	9.12D-01	0.0
0.0	-1.69D-01	-3.70D-02	-3.36D-01	2.71D-01	-6.93D-02	-1.09D-01	0.0
0.0	2.49D-01	2.01D-02	-5.39D-02	-6.93D-02	4.69D-01	5.66D-02	0.0
0.0	-7.00D-02	-1.01D-02	9.12D-01	-1.09D-01	5.66D-02	9.31D-01	0.0
-6.05D-01	0.0	0.0	0.0	0.0	0.0	0.0	6.05D-01

--- STF ---

2.36D 05	0.0	0.0	0.0	0.0	0.0	0.0	-2.36D 05
0.0	2.29D 06	1.15D 06	0.0	0.0	1.15D 06	0.0	0.0
0.0	1.15D 06	7.63D 05	0.0	0.0	3.82D 05	0.0	0.0
0.0	0.0	0.0	1.39D 06	-5.09D 05	-5.09D 05	7.07D 05	0.0
0.0	0.0	0.0	-5.09D 05	5.09D 05	2.54D 05	0.0	0.0
0.0	1.15D 06	3.82D 05	-5.09D 05	2.54D 05	1.27D 06	0.0	0.0
0.0	0.0	0.0	7.07D 05	0.0	0.0	7.07D 05	0.0
-2.36D 05	0.0	0.0	0.0	0.0	0.0	0.0	2.36D 05

Figure 6.3 CONFIGURATION I - Mass, Damping, and Stiffness

```
CONFIGURATION AND MATRIX LAYER =  1

MATRICES ARE  8X8

- ( BMIV ) -

 8.52D 03  0.0        0.0        0.0        0.0        0.0        0.0       -4.23D 03
 0.0       4.99D 04  -2.66D 05  -4.90D 02  -1.14D 04  -1.31D 04  -2.34D 02   0.0
 0.0      -2.66D 05   1.97D 06   7.95D 03   1.89D 05   2.18D 05   3.94D 03   0.0
 0.0      -4.90D 02   7.95D 03   1.02D 04   4.24D 04   8.16D 03   4.96D 03   0.0
 0.0      -1.14D 04   1.89D 05   4.24D 04   4.40D 05   1.97D 05   2.09D 04   0.0
 0.0      -1.31D 04   2.18D 05   8.16D 03   1.97D 05   2.30D 05   4.04D 03   0.0
 0.0      -2.34D 02   3.94D 03   4.96D 03   2.09D 04   4.04D 03   2.15D 04   0.0
-4.23D 03  0.0        0.0        0.0        0.0        0.0        0.0        8.52D 03

- ( DA ) -

 7.72D 03  0.0        0.0        0.0        0.0        0.0        0.0       -7.72D 03
 0.0      -1.35D 04  -2.31D 04   4.56D 03  -6.00D 02   1.73D 03  -9.68D 02   0.0
 0.0       2.53D 05   2.01D 05  -8.87D 03   4.95D 03   6.25D 04   1.29D 03   0.0
 0.0       1.81D 03   5.51D 02   5.75D 03   6.77D 03   6.55D 02   9.68D 03   0.0
 0.0       3.66D 04   1.52D 04  -6.27D 04   8.42D 04   6.15D 04   2.01D 04   0.0
 0.0       7.47D 04   2.53D 04  -4.78D 04   2.83D 04   9.52D 04   1.49D 03   0.0
 0.0      -4.46D 02   7.92D 01   2.01D 04   1.26D 03   1.42D 03   2.25D 04   0.0
-7.72D 03  0.0        0.0        0.0        0.0        0.0        0.0        7.72D 03

- ( DV ) -

 3.00D 09  0.0        0.0        0.0        0.0        0.0        0.0       -3.00D 09
 0.0      -2.05D 11  -1.51D 11   1.16D 10  -8.87D 09  -6.36D 10  -5.12D 08   0.0
 0.0       1.90D 12   1.28D 12  -1.93D 11   1.47D 11   7.70D 11   8.40D 09   0.0
 0.0       1.73D 10   8.62D 09  -8.15D 09   1.85D 10   1.85D 10   1.07D 10   0.0
 0.0       4.15D 11   2.05D 11  -2.51D 11   2.53D 11   4.00D 11   4.48D 10   0.0
 0.0       4.83D 11   2.39D 11  -2.03D 11   1.54D 11   4.07D 11   8.62D 09   0.0
 0.0       8.59D 09   4.28D 09   9.38D 09   9.13D 09   9.16D 09   1.87D 10   0.0
-3.00D 09  0.0        0.0        0.0        0.0        0.0        0.0        3.00D 09
```

Figure 6.4 CONFIGURATION I - Precalculated Matrices; BMIV, DA, and DV

CONFIGURATION AND MATRIX LAYER = 2

MATRICES ARE 7X7

--- SYSM ---

1.55D-04	0.0	0.0	0.0	0.0	0.0	7.77D-05
0.0	7.70D-05	1.09D-05	0.0	0.0	-6.42D-06	0.0
0.0	1.09D-05	1.97D-06	0.0	0.0	-1.48D-06	0.0
0.0	0.0	0.0	2.01D-04	-2.44D-05	1.44D-05	0.0
0.0	0.0	0.0	-2.44D-05	6.66D-06	-5.00D-06	0.0
0.0	-6.42D-06	-1.48D-06	1.44D-05	-5.00D-06	8.64D-06	0.0
7.77D-05	0.0	0.0	0.0	0.0	0.0	1.55D-04

--- DAMP ---

6.05D-01	0.0	0.0	0.0	0.0	0.0	-6.05D-01
0.0	1.32D 00	2.93D-01	5.54D-01	-1.69D-01	2.49D-01	0.0
0.0	2.93D-01	1.42D-01	1.00D-01	-3.70D-02	2.01D-02	0.0
0.0	5.54D-01	1.00D-01	1.51D 00	-3.36D-01	-5.39D-02	0.0
0.0	-1.69D-01	-3.70D-02	-3.36D-01	2.71D-01	-6.93D-02	0.0
0.0	2.49D-01	2.01D-02	-5.39D-02	-6.93D-02	4.69D-01	0.0
-6.05D-01	0.0	0.0	0.0	0.0	0.0	6.05D-01

--- STF ---

2.36D 05	0.0	0.0	0.0	0.0	0.0	-2.36D 05
0.0	2.29D 06	1.15D 06	0.0	0.0	1.15D 06	0.0
0.0	1.15D 06	7.63D 05	0.0	0.0	3.82D 05	0.0
0.0	0.0	0.0	1.39D 06	-5.09D 05	-5.09D 05	0.0
0.0	0.0	0.0	-5.09D 05	5.09D 05	2.54D 05	0.0
0.0	1.15D 06	3.82D 05	-5.09D 05	2.54D 05	1.27D 06	0.0
-2.36D 05	0.0	0.0	0.0	0.0	0.0	2.36D 05

Figure 6.5 CONFIGURATION II - Mass, Damping, and Stiffness

(This page intentionally blank)

CONFIGURATION AND MATRIX LAYER = 2

MATRICES ARE 7X1

COLUMN MATRICES FOR CALCULATION OF DELTA P

COLUMN MASS	COLUMN DAMPING	COLUMN STIFFNESS
0.0	0.0	0.0
0.0	-7.00D-02	0.0
0.0	-1.01D-02	0.0
-2.59D-05	9.12D-01	7.07D 05
0.0	-1.09D-01	0.0
0.0	5.66D-02	0.0
0.0	0.0	0.0

Figure 6.6 CONFIGURATION II - Precalculated Matrices; COLM, COLC, and COLK

CONFIGURATION AND MATRIX LAYER = 2

MATRICES ARE 7X7

- (BMIV) -

8.52D 03	0.0	0.0	0.0	0.0	0.0	-4.23D 03
0.0	4.99D 04	-2.66D 05	-4.36D 02	-1.11D 04	-1.31D 04	0.0
0.0	-2.66D 05	1.97D 06	7.04D 03	1.85D 05	2.17D 05	0.0
0.0	-4.36D 02	7.04D 03	9.03D 03	3.76D 04	7.23D 03	0.0
0.0	-1.11D 04	1.85D 05	3.76D 04	4.20D 05	1.93D 05	0.0
0.0	-1.31D 04	2.17D 05	7.23D 03	1.93D 05	2.29D 05	0.0
-4.23D 03	0.0	0.0	0.0	0.0	0.0	8.52D 03

- (DA) -

7.72D 03	0.0	0.0	0.0	0.0	0.0	-7.72D 03
0.0	-1.35D 04	-2.31D 04	4.78D 03	-5.86D 02	1.74D 03	0.0
0.0	2.53D 05	2.01D 05	-1.26D 04	4.71D 03	6.22D 04	0.0
0.0	1.91D 03	5.32D 02	1.10D 03	6.48D 03	3.28D 02	0.0
0.0	3.70D 04	1.51D 04	-8.22D 04	8.30D 04	6.01D 04	0.0
0.0	7.48D 04	2.53D 04	-5.16D 04	2.81D 04	9.49D 04	0.0
-7.72D 03	0.0	0.0	0.0	0.0	0.0	7.72D 03

- (DV) -

3.00D 09	0.0	0.0	0.0	0.0	0.0	-3.00D 09
0.0	-2.05D 11	-1.51D 11	1.17D 10	-8.77D 09	-6.35D 10	0.0
0.0	1.90D 12	1.28D 12	-1.95D 11	1.46D 11	7.68D 11	0.0
0.0	1.53D 10	7.64D 09	-1.03D 10	1.64D 10	1.64D 10	0.0
0.0	4.07D 11	2.02D 11	-2.60D 11	2.44D 11	3.91D 11	0.0
0.0	4.82D 11	2.39D 11	-2.05D 11	1.53D 11	4.05D 11	0.0
-3.00D 09	0.0	0.0	0.0	0.0	0.0	3.00D 09

Figure 6.7 CONFIGURATION II - Precalculated Matrices; BMIV, DA, and DV

CONFIGURATION AND MATRIX LAYER = 3

MATRICES ARE 7X7

--- SYSM ---

1.55D-04	7.77D-05	0.0	0.0	0.0	0.0	0.0
7.77D-05	2.32D-04	1.09D-05	0.0	0.0	-6.42D-06	0.0
0.0	1.09D-05	1.97D-06	0.0	0.0	-1.48D-06	0.0
0.0	0.0	0.0	2.01D-04	-2.44D-05	1.44D-05	-2.59D-05
0.0	0.0	0.0	-2.44D-05	6.66D-06	-5.00D-06	0.0
0.0	-6.42D-06	-1.48D-06	1.44D-05	-5.00D-06	8.64D-06	0.0
0.0	0.0	0.0	-2.59D-05	0.0	0.0	5.18D-05

--- DAMP ---

2.48D 00	1.05D 00	-7.37D-03	1.72D 00	-1.60D-01	9.80D-02	-6.62D-01
1.05D 00	3.49D 00	3.58D-01	2.09D 00	-3.08D-01	3.16D-01	-6.69D-01
-7.37D-03	3.58D-01	2.03D-01	1.02D-01	-3.48D-02	2.59D-03	-1.65D-02
1.72D 00	2.09D 00	1.02D-01	3.10D 00	-4.83D-01	3.25D-02	2.99D-01
-1.60D-01	-3.08D-01	-3.48D-02	-4.83D-01	2.86D-01	-8.02D-02	-5.40D-02
9.80D-02	3.16D-01	2.59D-03	3.25D-02	-8.02D-02	4.91D-01	2.90D-02
-6.62D-01	-6.69D-01	-1.65D-02	2.99D-01	-5.40D-02	2.90D-02	1.17D 00

--- STF ---

2.36D 05	-2.36D 05	0.0	0.0	0.0	0.0	0.0
-2.36D 05	2.53D 06	1.15D 06	0.0	0.0	1.15D 06	0.0
0.0	1.15D 06	7.63D 05	0.0	0.0	3.82D 05	0.0
0.0	0.0	0.0	1.39D 06	-5.09D 05	-5.09D 05	7.07D 05
0.0	0.0	0.0	-5.09D 05	5.09D 05	2.54D 05	0.0
0.0	1.15D 06	3.82D 05	-5.09D 05	2.54D 05	1.27D 06	0.0
0.0	0.0	0.0	7.07D 05	0.0	0.0	7.07D 05

Figure 6.8 CONFIGURATION III - Mass, Damping, and Stiffness

CONFIGURATION AND MATRIX LAYER = 3

MATRICES ARE 7X7

- (BMIV) -

8.18D 03	-3.62D 03	1.90D 04	1.34D 01	7.67D 02	9.06D 02	3.82D 01
-3.62D 03	7.24D 03	-3.81D 04	-7.89D 01	-1.58D 03	-1.82D 03	-2.14D 01
1.90D 04	-3.81D 04	7.50D 05	5.68D 03	1.35D 05	1.56D 05	2.79D 03
1.34D 01	-7.89D 01	5.68D 03	1.01D 04	4.23D 04	8.03D 03	4.97D 03
7.67D 02	-1.58D 03	1.35D 05	4.23D 04	4.38D 05	1.94D 05	2.08D 04
9.06D 02	-1.82D 03	1.56D 05	8.03D 03	1.94D 05	2.27D 05	3.98D 03
3.82D 01	-2.14D 01	2.79D 03	4.97D 03	2.08D 04	3.98D 03	2.15D 04

- (DA) -

1.63D 04	2.88D 03	2.48D 03	8.18D 03	-7.17D 02	9.49D 01	-3.28D 03
-1.09D 03	7.55D 03	-5.07D 03	5.46D 03	-5.96D 02	1.06D 03	-1.84D 03
3.14D 03	1.73D 05	1.35D 05	-1.18D 04	5.78D 03	5.79D 04	2.73D 03
8.08D 03	9.12D 03	6.32D 02	1.32D 04	6.10D 03	1.01D 03	6.76D 03
7.20D 03	4.42D 04	1.61D 04	-5.59D 04	8.36D 04	6.17D 04	1.74D 04
1.65D 03	7.66D 04	2.57D 04	-4.63D 04	2.80D 04	9.60D 04	1.22D 03
-8.56D 03	-8.19D 03	-4.62D-01	1.22D 04	1.98D 03	1.07D 03	2.56D 04

- (DV) -

2.78D 09	1.18D 10	1.07D 10	-8.06D 08	6.14D 08	4.46D 09	3.65D 07
-2.56D 09	-2.66D 10	-2.15D 10	1.61D 09	-1.23D 09	-8.93D 09	-7.09D 07
1.35D 10	9.37D 11	5.88D 11	-1.38D 11	1.06D 11	4.73D 11	5.99D 09
2.17D 07	1.55D 10	7.31D 09	-8.05D 09	1.84D 10	1.79D 10	1.07D 10
5.54D 08	3.72D 11	1.75D 11	-2.48D 11	2.51D 11	3.86D 11	4.46D 10
6.43D 08	4.34D 11	2.04D 11	-2.00D 11	1.52D 11	3.91D 11	8.49D 09
1.41D 07	7.69D 09	3.63D 09	9.44D 09	9.09D 09	8.87D 09	1.87D 10

Figure 6.9 CONFIGURATION III - Precalculated Matrices; BMIV, DA, and DV

CONFIGURATION AND MATRIX LAYER = 4

MATRICES ARE 6X6

--- SYSM ---

1.55D-04	7.77D-05	0.0	0.0	0.0	0.0
7.77D-05	2.32D-04	1.09D-05	0.0	0.0	-6.42D-06
0.0	1.09D-05	1.97D-06	0.0	0.0	-1.48D-06
0.0	0.0	0.0	2.01D-04	-2.44D-05	1.44D-05
0.0	0.0	0.0	-2.44D-05	6.66D-06	-5.00D-06
0.0	-6.42D-06	-1.48D-06	1.44D-05	-5.00D-06	8.64D-06

--- DAMP ---

2.48D 00	1.05D 00	-7.37D-03	1.72D 00	-1.60D-01	9.80D-02
1.05D 00	3.49D 00	3.58D-01	2.09D 00	-3.08D-01	3.16D-01
-7.37D-03	3.58D-01	2.03D-01	1.02D-01	-3.48D-02	2.59D-03
1.72D 00	2.09D 00	1.02D-01	3.10D 00	-4.83D-01	3.25D-02
-1.60D-01	-3.08D-01	-3.48D-02	-4.83D-01	2.86D-01	-8.02D-02
9.80D-02	3.16D-01	2.59D-03	3.25D-02	-8.02D-02	4.91D-01

--- STF ---

2.36D 05	-2.36D 05	0.0	0.0	0.0	0.0
-2.36D 05	2.53D 06	1.15D 06	0.0	0.0	1.15D 06
0.0	1.15D 06	7.63D 05	0.0	0.0	3.82D 05
0.0	0.0	0.0	1.39D 06	-5.09D 05	-5.09D 05
0.0	0.0	0.0	-5.09D 05	5.09D 05	2.54D 05
0.0	1.15D 06	3.82D 05	-5.09D 05	2.54D 05	1.27D 06

Figure 6.10 CONFIGURATION IV - Mass, Damping, and Stiffness

CONFIGURATION AND MATRIX LAYER = 4

MATRICES ARE 6X1

COLUMN MATRICES FOR CALCULATION OF DELTA P

COLUMN MASS	COLUMN DAMPING	COLUMN STIFFNESS
0.0	-6.62D-01	0.0
0.0	-6.69D-01	0.0
0.0	-1.65D-02	0.0
-2.59D-05	2.99D-01	7.07D 05
0.0	-5.40D-02	0.0
0.0	2.90D-02	0.0

Figure 6.11 CONFIGURATION IV - Precalculated Matrices; COLM, COLC, and COLK

CONFIGURATION AND MATRIX LAYER = 4

MATRICES ARE 6X6

- (BMIV) -

8.18D 03	-3.62D 03	1.90D 04	4.55D 00	7.30D 02	8.99D 02
-3.62D 03	7.24D 03	-3.81D 04	-7.39D 01	-1.56D 03	-1.82D 03
1.90D 04	-3.81D 04	7.49D 05	5.03D 03	1.32D 05	1.56D 05
4.55D 00	-7.39D 01	5.03D 03	9.00D 03	3.75D 04	7.11D 03
7.30D 02	-1.56D 03	1.32D 05	3.75D 04	4.18D 05	1.90D 05
8.99D 02	-1.82D 03	1.56D 05	7.11D 03	1.90D 05	2.26D 05

- (DA) -

1.63D 04	2.90D 03	2.48D 03	8.16D 03	-7.20D 02	9.30D 01
-1.10D 03	7.55D 03	-5.07D 03	5.47D 03	-5.94D 02	1.06D 03
4.26D 03	1.74D 05	1.35D 05	-1.33D 04	5.52D 03	5.78D 04
1.01D 04	1.10D 04	6.32D 02	1.04D 04	5.64D 03	7.64D 02
1.55D 04	5.21D 04	1.61D 04	-6.78D 04	8.17D 04	6.07D 04
3.24D 03	7.81D 04	2.57D 04	-4.86D 04	2.76D 04	9.58D 04

- (DV) -

2.78D 09	1.18D 10	1.07D 10	-8.23D 08	5.98D 08	4.45D 09
-2.56D 09	-2.66D 10	-2.15D 10	1.62D 09	-1.22D 09	-8.93D 09
1.35D 10	9.36D 11	5.88D 11	-1.40D 11	1.04D 11	4.72D 11
1.85D 07	1.37D 10	6.47D 09	-1.02D 10	1.63D 10	1.58D 10
5.40D 08	3.65D 11	1.72D 11	-2.57D 11	2.42D 11	3.77D 11
6.41D 08	4.32D 11	2.03D 11	-2.02D 11	1.50D 11	3.90D 11

Figure 6.12 CONFIGURATION IV - Precalculated Matrices; BMIV, DA, and DV

CONFIGURATION AND MATRIX LAYER = 5

MATRICES ARE 7X7

--- SYSM ---

7.70D-05	1.09D-05	0.0	0.0	-6.42D-06	0.0	0.0
1.09D-05	1.97D-06	0.0	0.0	-1.48D-06	0.0	0.0
0.0	0.0	2.01D-04	-2.44D-05	1.44D-05	-2.59D-05	0.0
0.0	0.0	-2.44D-05	6.66D-06	-5.00D-06	0.0	0.0
-6.42D-06	-1.48D-06	1.44D-05	-5.00D-06	8.64D-06	0.0	0.0
0.0	0.0	-2.59D-05	0.0	0.0	5.18D-05	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.55D-04

--- DAMP ---

1.32D 00	2.93D-01	5.54D-01	-1.69D-01	2.49D-01	-7.00D-02	0.0
2.93D-01	1.42D-01	1.00D-01	-3.70D-02	2.01D-02	-1.01D-02	0.0
5.54D-01	1.00D-01	1.51D 00	-3.36D-01	-5.39D-02	9.12D-01	0.0
-1.69D-01	-3.70D-02	-3.36D-01	2.71D-01	-6.93D-02	-1.09D-01	0.0
2.49D-01	2.01D-02	-5.39D-02	-6.93D-02	4.69D-01	5.66D-02	0.0
-7.00D-02	-1.01D-02	9.12D-01	-1.09D-01	5.66D-02	9.31D-01	0.0
0.0	0.0	0.0	0.0	0.0	0.0	6.05D-01

--- STF ---

2.29D 06	1.15D 06	0.0	0.0	1.15D 06	0.0	0.0
1.15D 06	7.63D 05	0.0	0.0	3.82D 05	0.0	0.0
0.0	0.0	1.39D 06	-5.09D 05	-5.09D 05	7.07D 05	0.0
0.0	0.0	-5.09D 05	5.09D 05	2.54D 05	0.0	0.0
1.15D 06	3.82D 05	-5.09D 05	2.54D 05	1.27D 06	0.0	0.0
0.0	0.0	7.07D 05	0.0	0.0	7.07D 05	0.0
0.0	0.0	0.0	0.0	0.0	0.0	2.36D 05

Figure 6.13 CONFIGURATION V - Mass, Damping, and Stiffness

CONFIGURATION AND MATRIX LAYER = 5

MATRICES ARE 7X1

COLUMN MATRICES FOR CALCULATION OF DELTA P

COLUMN MASS	COLUMN DAMPING	COLUMN STIFFNESS
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
7.77D-05	-6.05D-01	-2.36D 05

Figure 6.14 CONFIGURATION V - Precalculated Matrices; COLM, COLC, and COLK

CONFIGURATION AND MATRIX LAYER = 5

MATRICES ARE 7X7

- (BMIV) -

4.99D 04	-2.66D 05	-4.90D 02	-1.14D 04	-1.31D 04	-2.34D 02	0.0
-2.66D 05	1.97D 06	7.95D 03	1.89D 05	2.18D 05	3.94D 03	0.0
-4.90D 02	7.95D 03	1.02D 04	4.24D 04	8.16D 03	4.96D 03	0.0
-1.14D 04	1.89D 05	4.24D 04	4.40D 05	1.97D 05	2.09D 04	0.0
-1.31D 04	2.18D 05	8.16D 03	1.97D 05	2.30D 05	4.04D 03	0.0
-2.34D 02	3.94D 03	4.96D 03	2.09D 04	4.04D 03	2.15D 04	0.0
0.0	0.0	0.0	0.0	0.0	0.0	6.42D 03

- (DA) -

-1.35D 04	-2.31D 04	4.56D 03	-6.00D 02	1.73D 03	-9.68D 02	0.0
2.53D 05	2.01D 05	-8.87D 03	4.95D 03	6.25D 04	1.29D 03	0.0
1.81D 03	5.51D 02	5.75D 03	6.77D 03	6.55D 02	9.68D 03	0.0
3.66D 04	1.52D 04	-6.27D 04	8.42D 04	6.15D 04	2.01D 04	0.0
7.47D 04	2.53D 04	-4.78D 04	2.83D 04	9.52D 04	1.49D 03	0.0
-4.46D 02	7.92D 01	2.01D 04	1.26D 03	1.42D 03	2.25D 04	0.0
0.0	0.0	0.0	0.0	0.0	0.0	3.88D 03

- (DV) -

-2.05D 11	-1.51D 11	1.16D 10	-8.87D 09	-6.36D 10	-5.12D 08	0.0
1.90D 12	1.28D 12	-1.93D 11	1.47D 11	7.70D 11	8.40D 09	0.0
1.73D 10	8.62D 09	-8.15D 09	1.85D 10	1.85D 10	1.07D 10	0.0
4.15D 11	2.06D 11	-2.51D 11	2.53D 11	4.00D 11	4.48D 10	0.0
4.83D 11	2.39D 11	-2.03D 11	1.54D 11	4.07D 11	8.62D 09	0.0
8.59D 09	4.28D 09	9.38D 09	9.13D 09	9.16D 09	1.87D 10	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.51D 09

Figure 6.15 CONFIGURATION V - Precalculated Matrics; BMIV, DA, and DV

CONFIGURATION AND MATRIX LAYER = 6

MATRICES ARE 6X6

--- SYSM ---

7.70D-05	1.09D-05	0.0	0.0	-6.42D-06	0.0
1.09D-05	1.97D-06	0.0	0.0	-1.48D-06	0.0
0.0	0.0	2.01D-04	-2.44D-05	1.44D-05	0.0
0.0	0.0	-2.44D-05	6.66D-06	-5.00D-06	0.0
-6.42D-06	-1.48D-06	1.44D-05	-5.00D-06	8.64D-06	0.0
0.0	0.0	0.0	0.0	0.0	1.55D-04

--- DAMP ---

1.32D 00	2.93D-01	5.54D-01	-1.69D-01	2.49D-01	0.0
2.93D-01	1.42D-01	1.00D-01	-3.70D-02	2.01D-02	0.0
5.54D-01	1.00D-01	1.51D 00	-3.36D-01	-5.39D-02	0.0
-1.69D-01	-3.70D-02	-3.36D-01	2.71D-01	-6.93D-02	0.0
2.49D-01	2.01D-02	-5.39D-02	-6.93D-02	4.69D-01	0.0
0.0	0.0	0.0	0.0	0.0	6.05D-01

--- STF ---

2.29D 06	1.15D 06	0.0	0.0	1.15D 06	0.0
1.15D 06	7.63D 05	0.0	0.0	3.82D 05	0.0
0.0	0.0	1.39D 06	-5.09D 05	-5.09D 05	0.0
0.0	0.0	-5.09D 05	5.09D 05	2.54D 05	0.0
1.15D 06	3.82D 05	-5.09D 05	2.54D 05	1.27D 06	0.0
0.0	0.0	0.0	0.0	0.0	2.36D 05

Figure 6.16 CONFIGURATION VI - Mass, Damping, and Stiffness

CONFIGURATION AND MATRIX LAYER = 6

MATRICES ARE 6X1

COLUMN MATRICES FOR CALCULATION OF DELTA P

COLUMN MASS	COLUMN DAMPING	COLUMN STIFFNESS
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
7.77D-05	-6.05D-01	-2.36D 05

Figure 6.17 CONFIGURATION VI - Precalculated Matrices; COLM, COLC, and COLK

CONFIGURATION = 6 MATRIX LAYER = 1

MATRICES ARE 6X1

COLUMN MATRICES FOR CALCULATION OF DELTA P

COLUMN MASS	COLUMN DAMPING	COLUMN STIFFNESS
0.0	-7.00D-02	0.0
0.0	-1.01D-02	0.0
-2.59D-05	9.12D-01	7.07D 05
0.0	-1.09D-01	0.0
0.0	5.66D-02	0.0
0.0	0.0	0.0

Figure 6.18 CONFIGURATION VI - Precalculated Matrices; COLM, COLC, and COLK

CONFIGURATION AND MATRIX LAYER = 6

MATRICES ARE 6X6

- (BMIV) -

4.99D 04	-2.66D 05	-4.36D 02	-1.11D 04	-1.31D 04	0.0
-2.66D 05	1.97D 06	7.04D 03	1.85D 05	2.17D 05	0.0
-4.36D 02	7.04D 03	9.03D 03	3.76D 04	7.23D 03	0.0
-1.11D 04	1.85D 05	3.76D 04	4.20D 05	1.93D 05	0.0
-1.31D 04	2.17D 05	7.23D 03	1.93D 05	2.29D 05	0.0
0.0	0.0	0.0	0.0	0.0	6.42D 03

- (DA) -

-1.35D 04	-2.31D 04	4.78D 03	-5.86D 02	1.74D 03	0.0
2.53D 05	2.01D 05	-1.26D 04	4.71D 03	6.22D 04	0.0
1.91D 03	5.32D 02	1.10D 03	6.48D 03	3.28D 02	0.0
3.70D 04	1.51D 04	-8.22D 04	8.30D 04	6.01D 04	0.0
7.48D 04	2.53D 04	-5.16D 04	2.81D 04	9.49D 04	0.0
0.0	0.0	0.0	0.0	0.0	3.88D 03

- (DV) -

-2.05D 11	-1.51D 11	1.17D 10	-8.77D 09	-6.35D 10	0.0
1.90D 12	1.28D 12	-1.95D 11	1.46D 11	7.68D 11	0.0
1.53D 10	7.64D 09	-1.03D 10	1.64D 10	1.64D 10	0.0
4.07D 11	2.02D 11	-2.60D 11	2.44D 11	3.91D 11	0.0
4.82D 11	2.39D 11	-2.05D 11	1.53D 11	4.05D 11	0.0
0.0	0.0	0.0	0.0	0.0	1.51D 09

Figure 6.19 CONFIGURATION VI - Precalculated Matrices; BMIV, DA, and DV

CONFIGURATION AND MATRIX LAYER = 7

MATRICES ARE 6X6

--- SYSM ---

2.32D-04	1.09D-05	0.0	0.0	-6.42D-06	0.0
1.09D-05	1.97D-06	0.0	0.0	-1.48D-06	0.0
0.0	0.0	2.01D-04	-2.44D-05	1.44D-05	-2.59D-05
0.0	0.0	-2.44D-05	6.66D-06	-5.00D-06	0.0
-6.42D-06	-1.48D-06	1.44D-05	-5.00D-06	8.64D-06	0.0
0.0	0.0	-2.59D-05	0.0	0.0	5.18D-05

--- DAMP ---

3.49D 00	3.58D-01	2.09D 00	-3.08D-01	3.16D-01	-6.69D-01
3.58D-01	2.03D-01	1.02D-01	-3.48D-02	2.59D-03	-1.65D-02
2.09D 00	1.02D-01	3.10D 00	-4.83D-01	3.25D-02	2.99D-01
-3.08D-01	-3.48D-02	-4.83D-01	2.86D-01	-8.02D-02	-5.40D-02
3.16D-01	2.59D-03	3.25D-02	-8.02D-02	4.91D-01	2.90D-02
-6.69D-01	-1.65D-02	2.99D-01	-5.40D-02	2.90D-02	1.17D 00

--- STF ---

2.53D 06	1.15D 06	0.0	0.0	1.15D 06	0.0
1.15D 06	7.63D 05	0.0	0.0	3.82D 05	0.0
0.0	0.0	1.39D 06	-5.09D 05	-5.09D 05	7.07D 05
0.0	0.0	-5.09D 05	5.09D 05	2.54D 05	0.0
1.15D 06	3.82D 05	-5.09D 05	2.54D 05	1.27D 06	0.0
0.0	0.0	7.07D 05	0.0	0.0	7.07D 05

Figure 6.20 CONFIGURATION VII - Mass, Damping, and Stiffness

CONFIGURATION AND MATRIX LAYER = 7

MATRICES ARE 6X1

COLUMN MATRICES FOR CALCULATION OF DELTA P

COLUMN MASS	COLUMN DAMPING	COLUMN STIFFNESS
7.77D-05	1.05D 00	-2.36D 05
0.0	-7.37D-03	0.0
0.0	1.72D 00	0.0
0.0	-1.60D-01	0.0
0.0	9.80D-02	0.0
0.0	-6.62D-01	0.0

Figure 6.21 CONFIGURATION VII - Precalculated Matrices; COLM, COLC, and COLK

CONFIGURATION AND MATRIX LAYER = 7

MATRICES ARE 6X6

- (BMIV) -

5.65D 03	-2.97D 04	-7.29D 01	-1.24D 03	-1.42D 03	-4.55D 00
-2.97D 04	7.05D 05	5.65D 03	1.33D 05	1.54D 05	2.71D 03
-7.29D 01	5.65D 03	1.01D 04	4.23D 04	8.03D 03	4.97D 03
-1.24D 03	1.33D 05	4.23D 04	4.38D 05	1.93D 05	2.08D 04
-1.42D 03	1.54D 05	8.03D 03	1.93D 05	2.27D 05	3.97D 03
-4.55D 00	2.71D 03	4.97D 03	2.08D 04	3.97D 03	2.15D 04

- (DA) -

8.83D 03	-3.98D 03	9.07D 03	-9.13D 02	1.10D 03	-3.29D 03
1.67D 05	1.29D 05	-3.08D 04	7.44D 03	5.77D 04	1.04D 04
9.12D 03	6.28D 02	1.32D 04	6.10D 03	1.01D 03	6.76D 03
4.39D 04	1.59D 04	-5.67D 04	8.37D 04	6.17D 04	1.77D 04
7.63D 04	2.54D 04	-4.72D 04	2.81D 04	9.60D 04	1.59D 03
-8.21D 03	-1.21D 01	1.22D 04	1.98D 03	1.07D 03	2.56D 04

- (DV) -

-2.14D 10	-1.68D 10	1.25D 09	-9.58D 08	-6.96D 09	-5.48D 07
9.09D 11	5.63D 11	-1.37D 11	1.04D 11	4.62D 11	5.91D 09
1.55D 10	7.29D 09	-8.04D 09	1.84D 10	1.79D 10	1.07D 10
3.71D 11	1.74D 11	-2.48D 11	2.50D 11	3.85D 11	4.46D 10
4.32D 11	2.03D 11	-2.00D 11	1.52D 11	3.91D 11	8.48D 09
7.64D 09	3.58D 09	9.44D 09	9.09D 09	8.85D 09	1.87D 10

Figure 6.22 CONFIGURATION VII - Precalculated Matrices; BMIV, DA, and DV

CONFIGURATION AND MATRIX LAYER = 8

MATRICES ARE 5X5

--- SYSM ---

2.32D-04	1.09D-05	0.0	0.0	-6.42D-06
1.09D-05	1.97D-06	0.0	0.0	-1.48D-06
0.0	0.0	2.01D-04	-2.44D-05	1.44D-05
0.0	0.0	-2.44D-05	6.66D-06	-5.00D-06
-6.42D-06	-1.48D-06	1.44D-05	-5.00D-06	8.64D-06

--- DAMP ---

3.49D 00	3.58D-01	2.09D 00	-3.08D-01	3.16D-01
3.58D-01	2.03D-01	1.02D-01	-3.48D-02	2.59D-03
2.09D 00	1.02D-01	3.10D 00	-4.83D-01	3.25D-02
-3.08D-01	-3.48D-02	-4.83D-01	2.86D-01	-8.02D-02
3.16D-01	2.59D-03	3.25D-02	-8.02D-02	4.91D-01

--- STF ---

2.53D 06	1.15D 06	0.0	0.0	1.15D 06
1.15D 06	7.63D 05	0.0	0.0	3.82D 05
0.0	0.0	1.39D 06	-5.09D 05	-5.09D 05
0.0	0.0	-5.09D 05	5.09D 05	2.54D 05
1.15D 06	3.82D 05	-5.09D 05	2.54D 05	1.27D 06

Figure 6.23 CONFIGURATION VIII - Mass, Damping, and Stiffness

CONFIGURATION AND MATRIX LAYER = 8

MATRICES ARE 5X1

COLUMN MATRICES FOR CALCULATION OF DELTA P

COLUMN MASS	COLUMN DAMPING	COLUMN STIFFNESS
7.77D-05	1.05D 00	-2.36D 05
0.0	-7.37D-03	0.0
0.0	1.72D 00	0.0
0.0	-1.60D-01	0.0
0.0	9.80D-02	0.0

CONFIGURATION = 8, MATRIX LAYER = 3

MATRICES ARE 5X1

COLUMN MATRICES FOR CALCULATION OF DELTA P

COLUMN MASS	COLUMN DAMPING	COLUMN STIFFNESS
0.0	-6.69D-01	0.0
0.0	-1.65D-02	0.0
-2.59D-05	2.99D-01	7.07D 05
0.0	-5.40D-02	0.0
0.0	2.90D-02	0.0

Figure 6.24 CONFIGURATION VIII - Precalculated Matrices; COLM, COLC, and COLK

CONFIGURATION AND MATRIX LAYER = 8

MATRICES ARE 5X5

- (BMIV) -

5.65D 03	-2.97D 04	-7.19D 01	-1.24D 03	-1.42D 03
-2.97D 04	7.05D 05	5.02D 03	1.31D 05	1.54D 05
-7.19D 01	5.02D 03	9.00D 03	3.75D 04	7.11D 03
-1.24D 03	1.31D 05	3.75D 04	4.18D 05	1.90D 05
-1.42D 03	1.54D 05	7.11D 03	1.90D C5	2.26D 05

- (DA) -

8.83D 03	-3.98D 03	9.07D 03	-9.12D 02	1.10D 03
1.68D 05	1.29D 05	-3.23D 04	7.19D 03	5.75D 04
1.10D 04	6.31D 02	1.04D 04	5.64D 03	7.64D 02
5.19D 04	1.59D 04	-6.85D 04	8.18D 04	6.07D 04
7.78D 04	2.54D 04	-4.95D 04	2.77D 04	9.58D 04

- (DV) -

-2.14D 10	-1.68D 10	1.26D 09	-9.56D 08	-6.96D 09
9.08D 11	5.63D 11	-1.38D 11	1.03D 11	4.61D 11
1.37D 10	6.47D 09	-1.02D 10	1.63D 10	1.58D 10
3.64D 11	1.71D 11	-2.57D 11	2.42D 11	3.77D 11
4.31D 11	2.02D 11	-2.02D 11	1.50D 11	3.89D 11

Figure 6.25 CONFIGURATION VIII - Precalculated Matrices; BMIV, DA, and DV

C. SAMPLE PROBLEMS

One set of results for 11,000 RPM was obtained in CALCOMP graph form and as AGT graphic display photographs. At this high speed of cam rotation, the pushrod is frequently thrown off the cam. The results are completely analyized with numerious reference points plotted on the graphs.

The second example problem was chosen at a slower speed, 9,000 RPM. At this speed the pushrod remains in contact with the cam. A third intermediate example, 10,000 RPM was chosen for a comparison between the two above examples. Figure 6.26 shows the relationship between the sample problems.

Photographs from the AGT display are presented after the CALCOMP graphs for each case of example one. It is noted that large scale factors are introduced for both bending and axial deformation as an aid to the user.

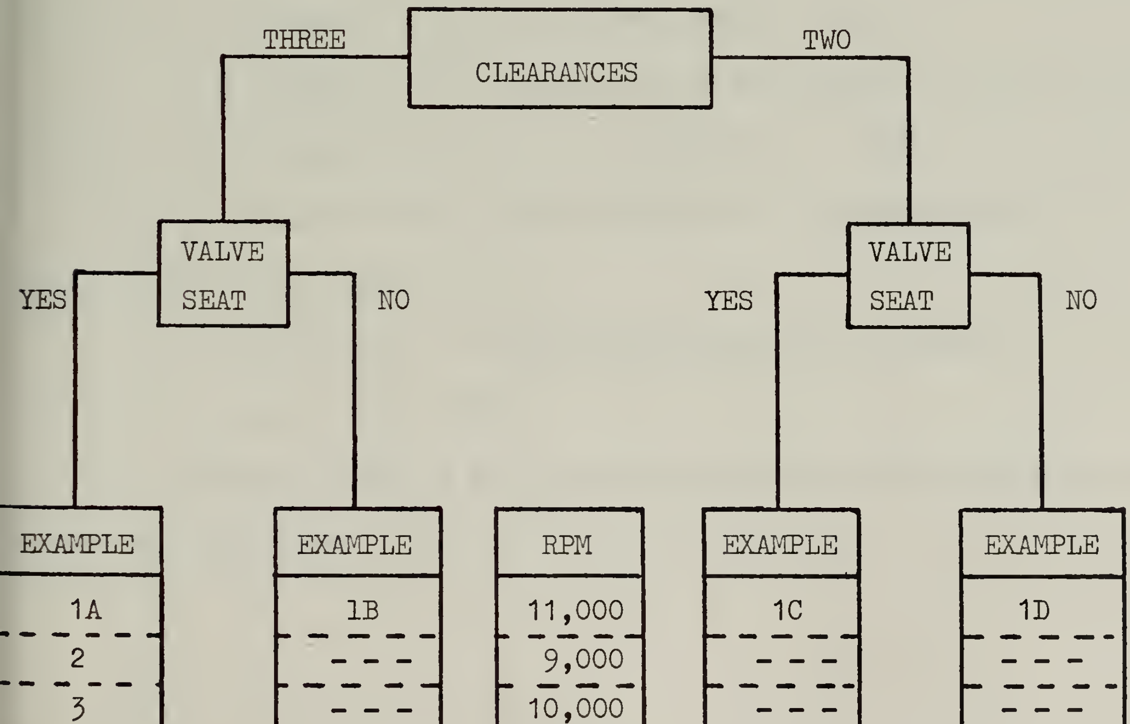


Figure 6.26 SAMPLE NUMERICAL RESULTS

1. <u>Example 1A, Three Clearance, 11,000 RPM, Valve Seat</u>

This analysis will commence at 270 degrees of cam rotation. One cycle will be discussed, from point (A) to point (AA). The following figures are utilized for the analysis:

a. Figure 6.27

(1) Displacement difference between cam and pushrod $(q_1 - q_9) \times 10^{-2}$

(2) Displacement of lower end of pushrod i.e. follower, $q_1 \times 10^{-1}$

(3) Cam Profile $q_9 \times 10^{-1}$

b. Figure 6.28

(1) Displacement difference between pushrod and rocker $(q_2 - q_8) \times 10^{-2}$

(2) Displacement difference between cam and pushrod $(q_1 - q_9) \times 10^{-2}$

c. Figure 6.29 Force on the Pushrod $F_1 \times 10^3$

d. Figure 6.30 Force on the Rocker $F_2 \times 10^2$

e. Figure 6.31 Displacement of the Valve $q_7 \times 10^{-1}$

f. Figure 6.32 Force on the Valve $F_7 \times 10^3$

g. Figure 6.33 Axial Oscillation of the Valve Stem $(q_7 - q_6) \times 10^{-3}$

h. Figure 6.34 Axial Oscillations of the Pushrod $(q_8 - q_1) \times 10^{-3}$

Figures 6.33 and 6.34 have been included for completeness and are not annotated.

The following reference points approximately correspond to the indicated degrees of cam rotation, Table 6.1

Reference Point	Degrees of Cam Rotation	Reference Point	Degrees of Cam Rotation
A	270	I	380
B	288	J	418
C	294	K	430
D	300	L	448
E	336	M	459
F	346	N	579
G	357	P	583
H	363	AA	630

Table 6.1 Graph Reference Points and Degrees of Cam Rotation for Figures 6.27 - 6.32

At point (A), the cam is at its lowest extreme and is starting to rise. The pushrod is floating between the cam and rocker, with downward motion. The valve has just undergone a severe oscillation and has resumed its upward motion.

At point (B), the cam and pushrod impact and remain in contact for a short period, (B-C). The rising cam and force of impact create severe oscillation of the pushrod and a bounce occurs. The pushrod is again thrown clear of the cam and floats (C-E).

At point (D), the valve has bounced against the valve seat, creating severe oscillation and more bouncing of the valve (D-E). The pushrod is floating thus is not affected by this impact.

At point (E), the rising pushrod now impacts against the rocker, and valve against the valve seat. At point (E), the valve has again reseated and the rocker has become almost stationary. This allows the rising pushrod to impact against the rocker.

The pushrod bounces off the rocker (F-H) and again floats (F-H), now having downward motion and also creating downward motion of the valve.

The rising cam almost immediately contacts the pushrod (G). The resulting motion creates positive contact between cam, pushrod, and rocker (H-I), and the valve bounces cease and it resumes its downward harmonic motion. At point (I), the speed of the cam throws the pushrod clear. Immediately separation occurs at the rocker- pushrod due to inertial effects. Several bounces at this clearance are observed (J-K-L-M), until the system reaches its extreme, point (M), i.e. the cam now with downward motion begins falling away from the pushrod. The valve spring, having reached the maximum extension, reverses the motion of the valve upward. The rocker and pushrod are now in contact (M) and both have downward motion.

At point (N), the valve impacts against the seat which results in severe oscillations and valve bounce (P-AA). With the gross motion of the rocker halted, the pushrod floats clear (P-AA). The cycle then repeats.

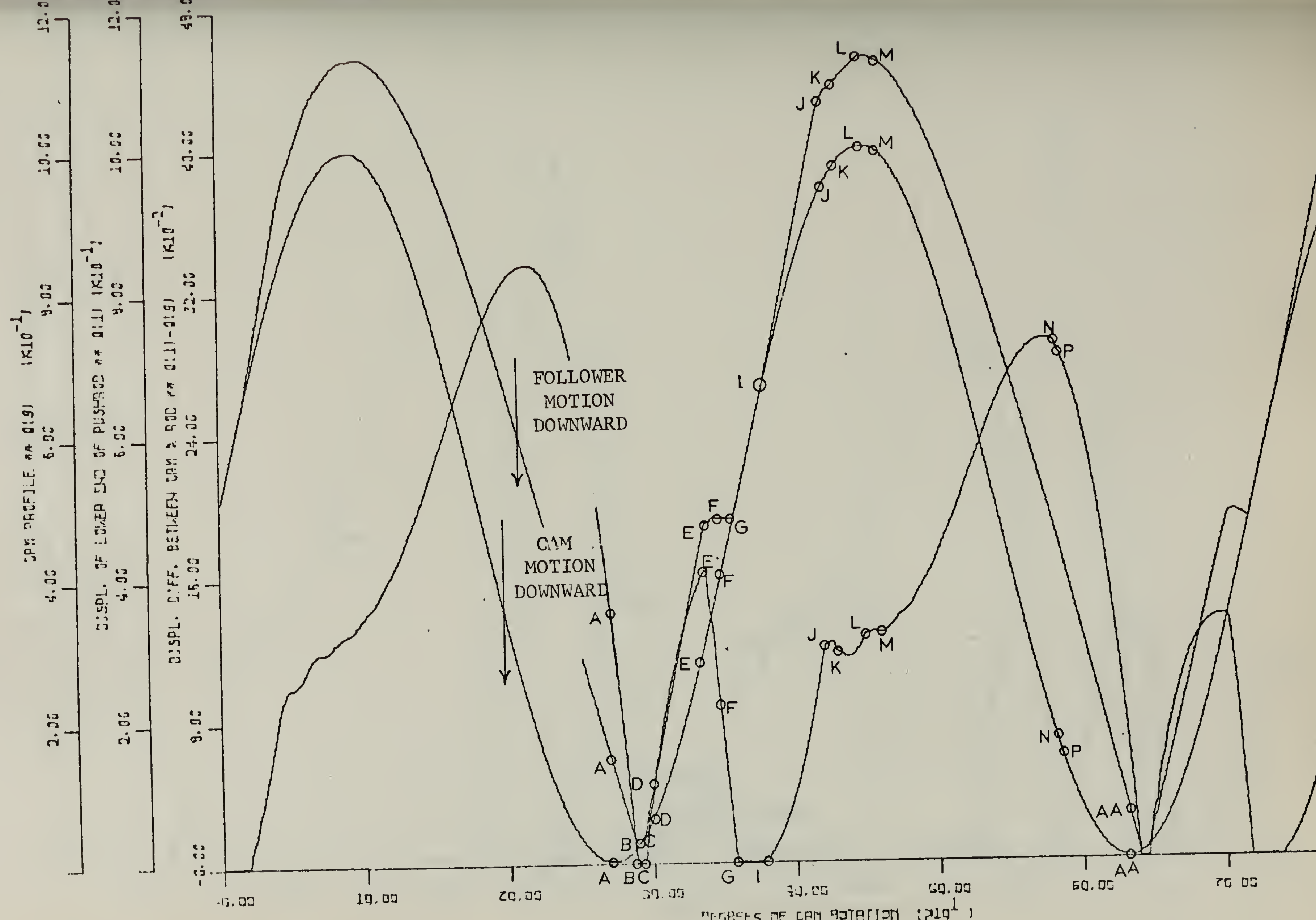


Figure 6.27 THREE CLEARANCES 11,000 RPM VALVE SEAT

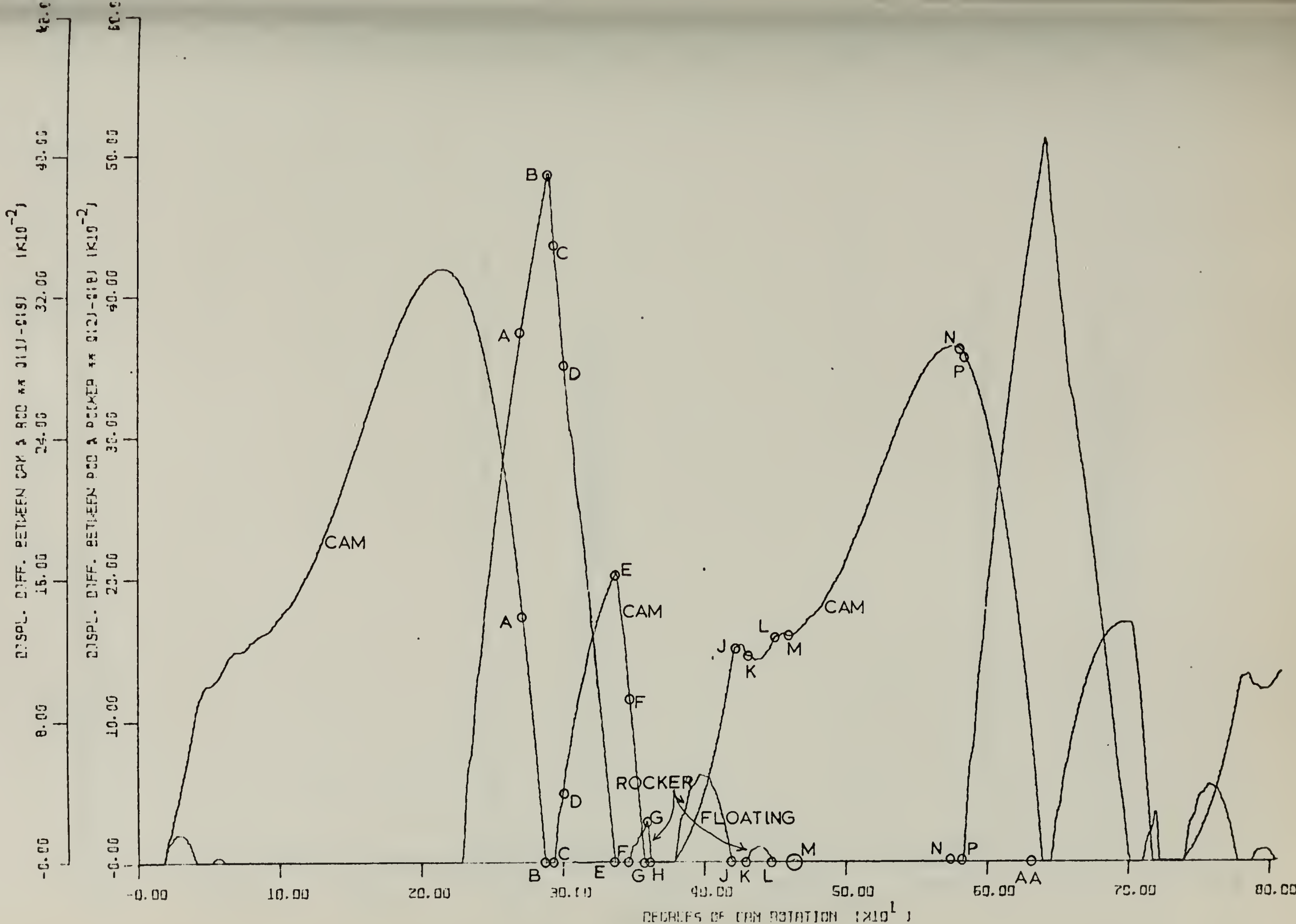


Figure 6.28 THREE CLEARANCES 11,000 RPM VALVE SEAT

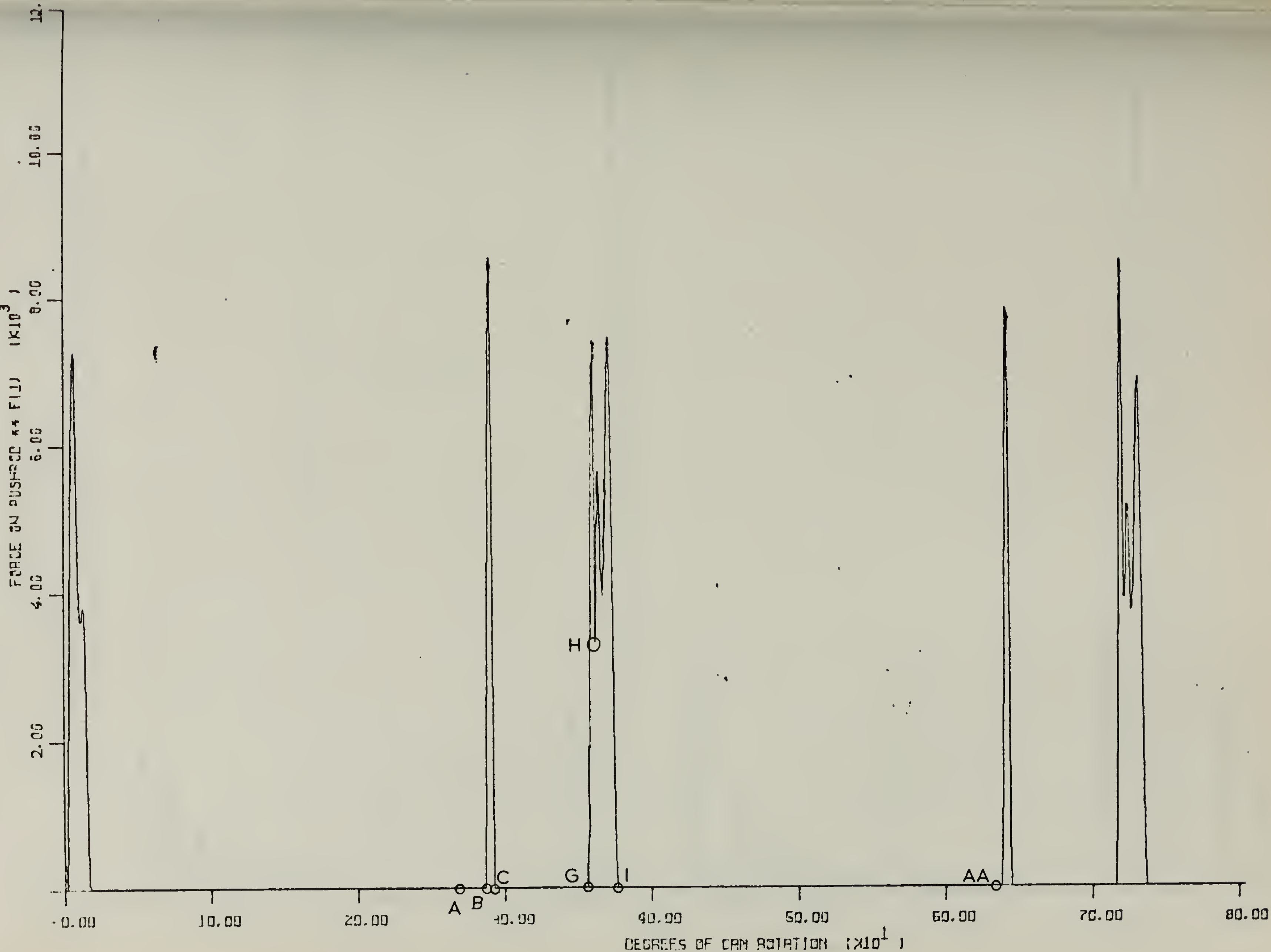


Figure 6.29 THREE CLEARANCES 11,000 RPM VALVE SEAT

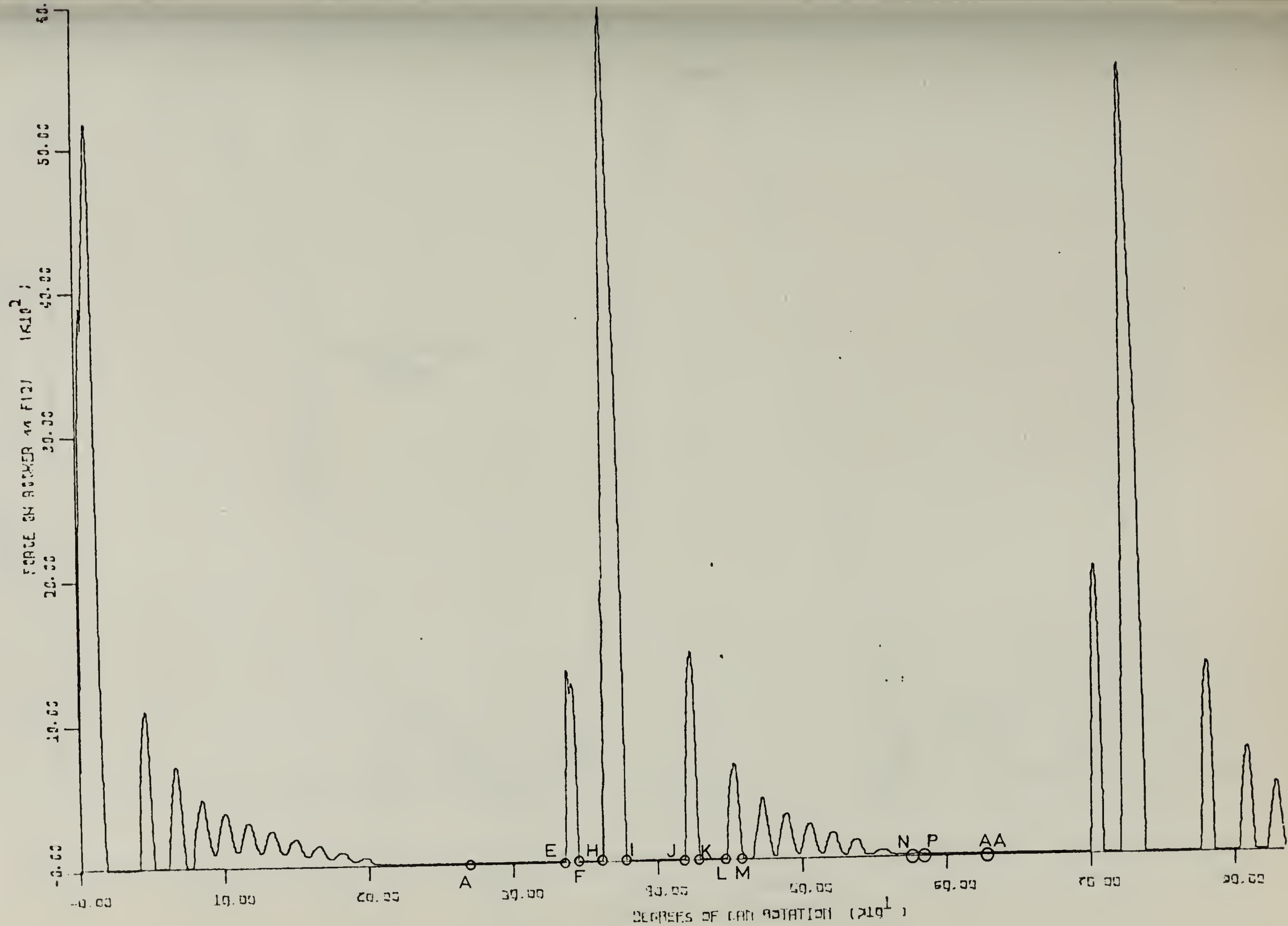


Figure 6.30 THREE CLEARANCES 11,000 RPM VALVE SEAT

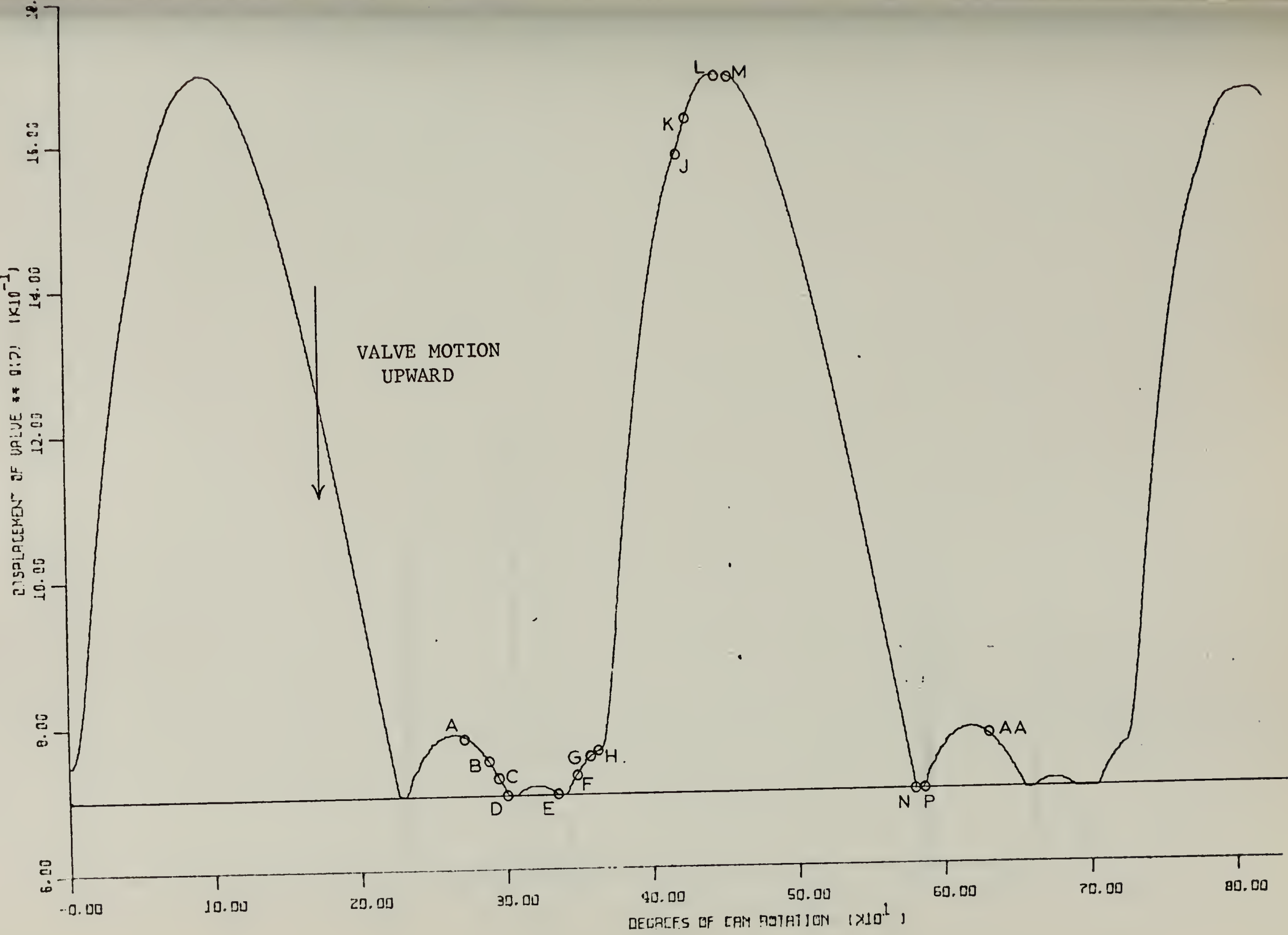


Figure 6.31 THREE CLEARANCES 11,000 RPM VALVE SEAT

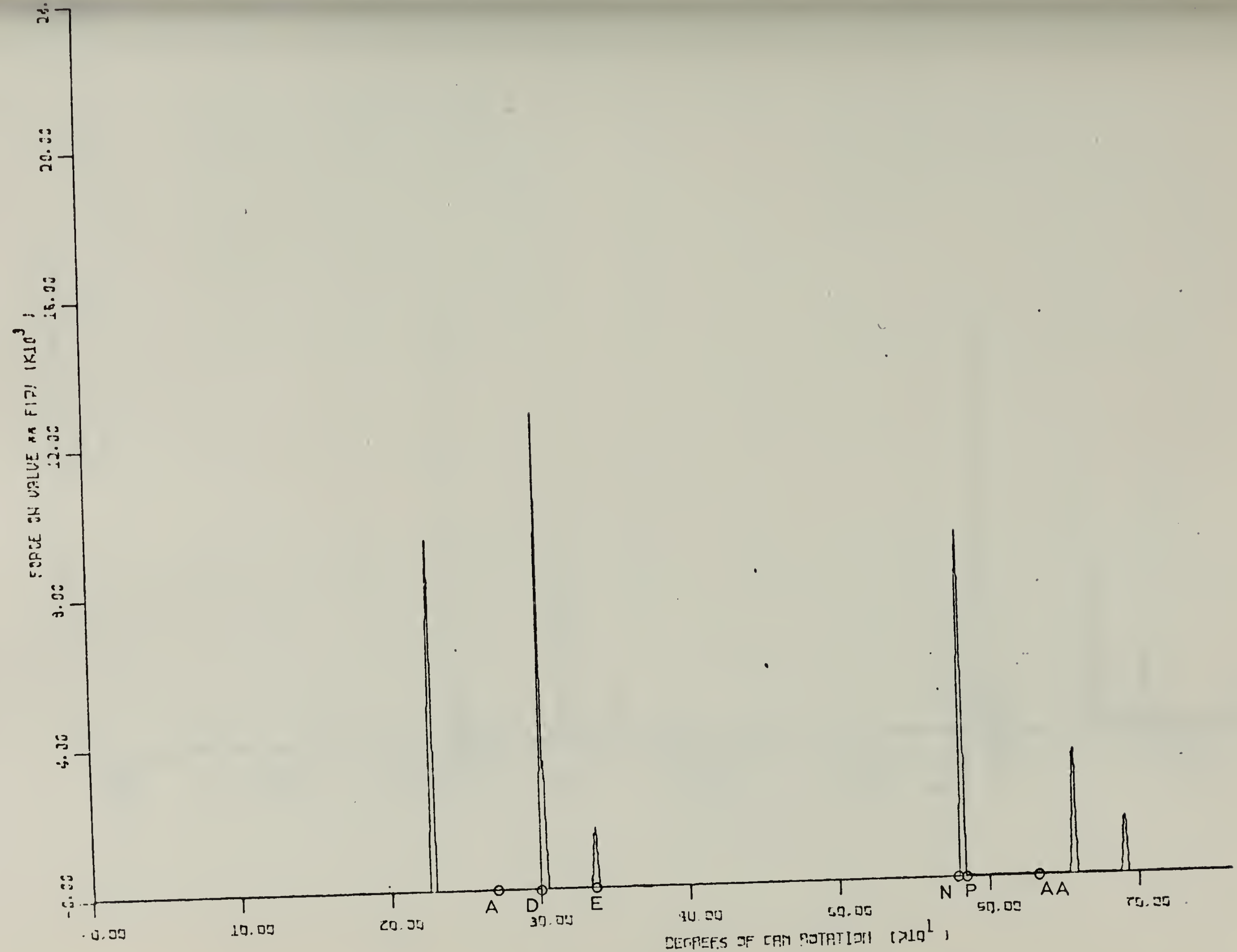


Figure 6.32 THREE CLEARANCES 11,000 RPM VALVE SEAT

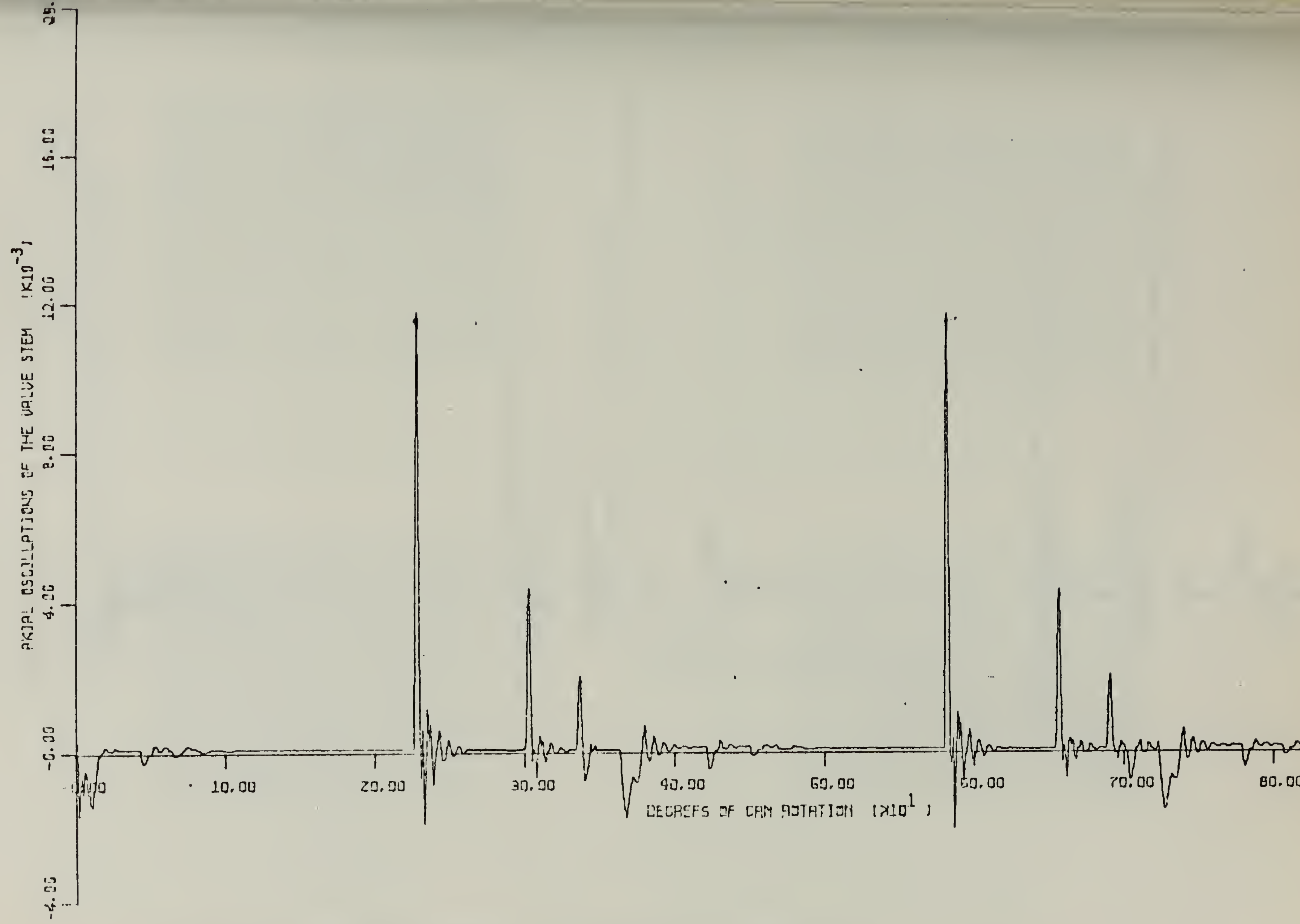


Figure 6.33 THREE CLEARANCES 11,000 RPM VALVE SEAT

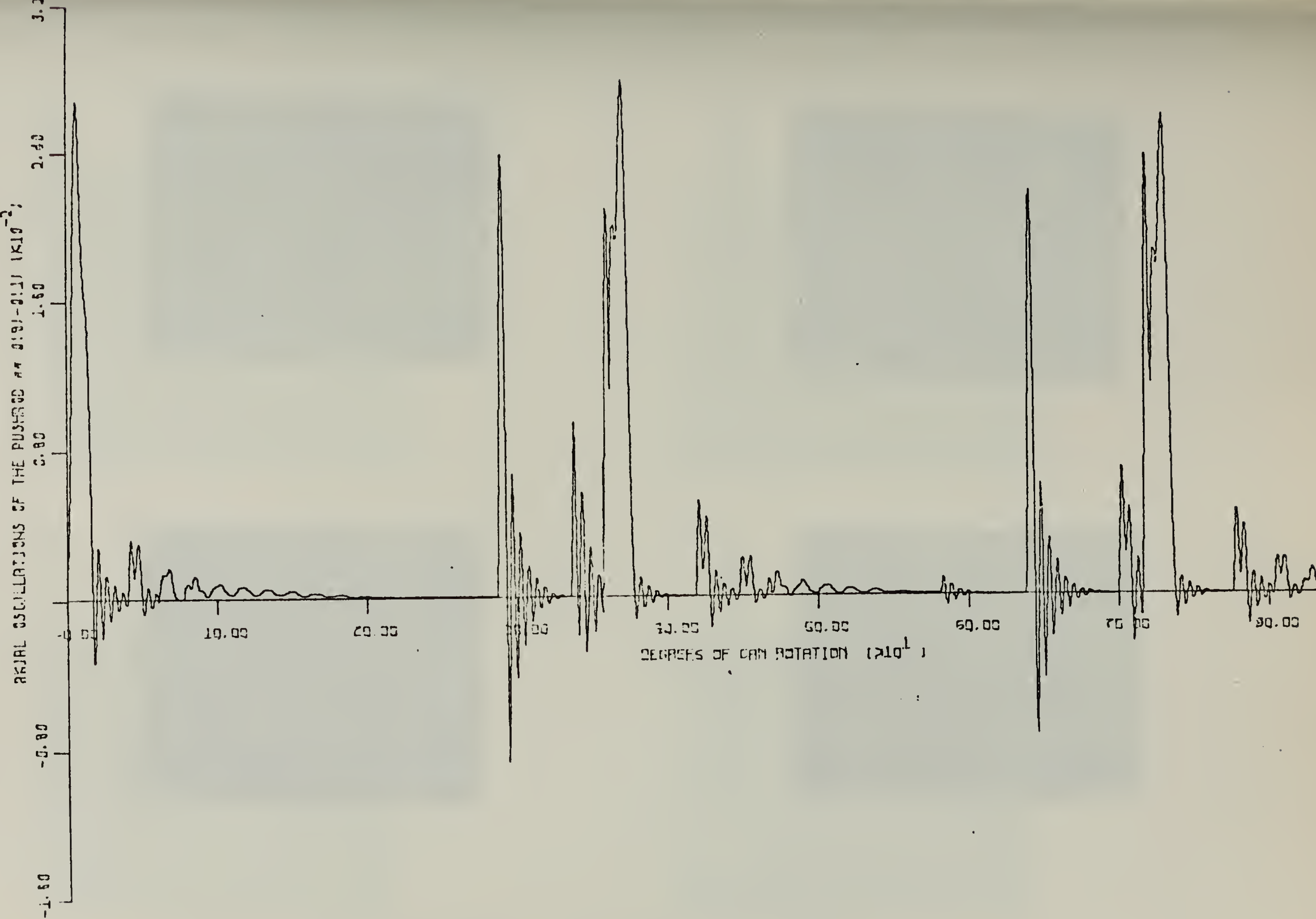


Figure 6.34 THREE CLEARANCES 11,000 RPM VALVE SEAT

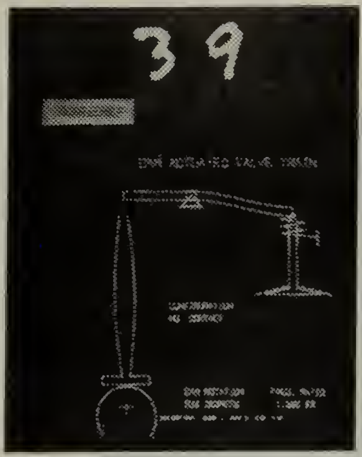


Photo 6.2 - 630 degrees

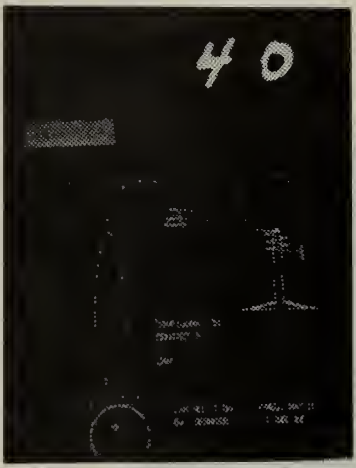


Photo 6.3 - 641 degrees

Photo 6.4 - 650 degrees

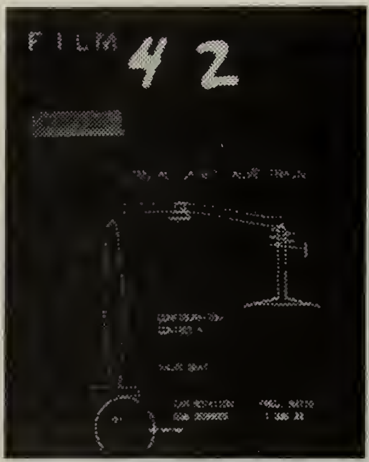


Photo 6.5 - 658 degrees

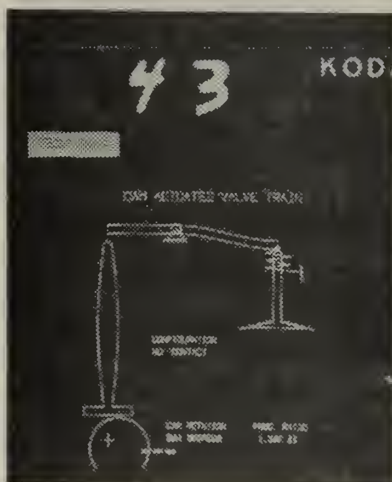


Photo 6.6 - 664 degrees

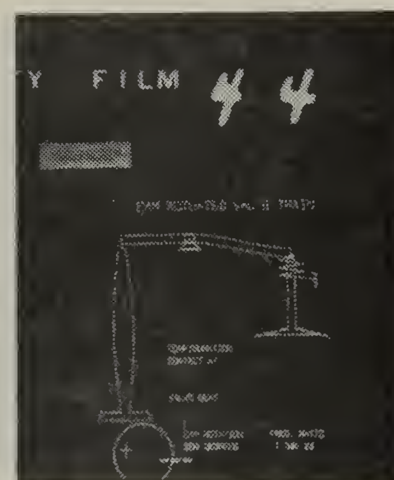


Photo 6.7 - 682 degrees

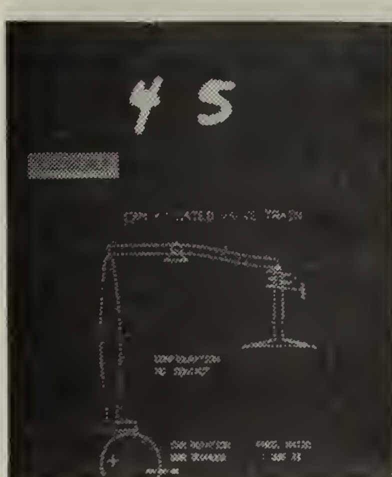


Photo 6.8 - 686 degrees

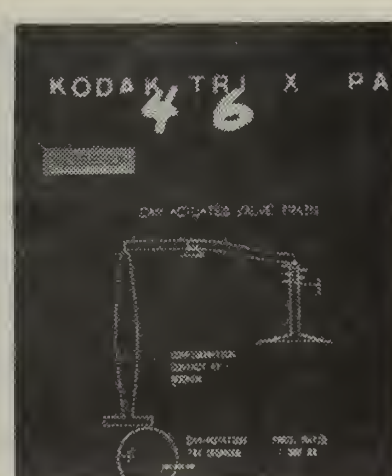


Photo 6.9 - 704 degrees

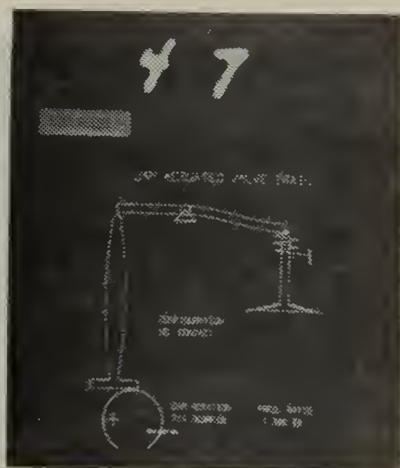


Photo 6.10 - 714 degrees

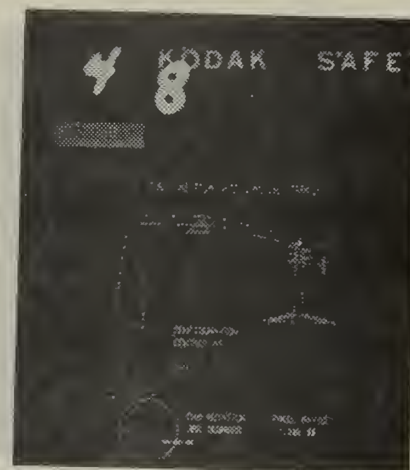


Photo 6.11 - 362 degrees

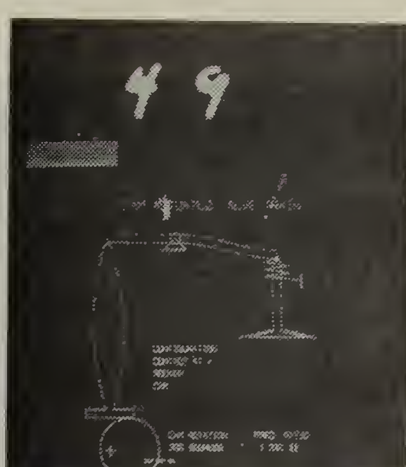


Photo 6.12 - 366 degrees

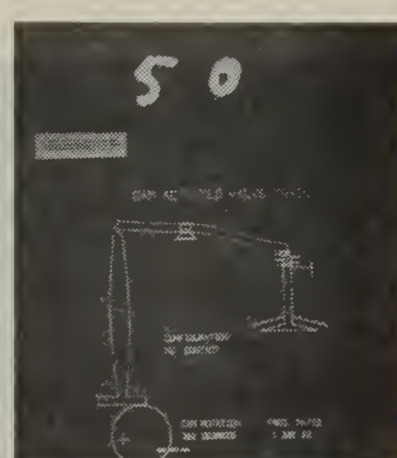


Photo 6.13 - 380 degrees

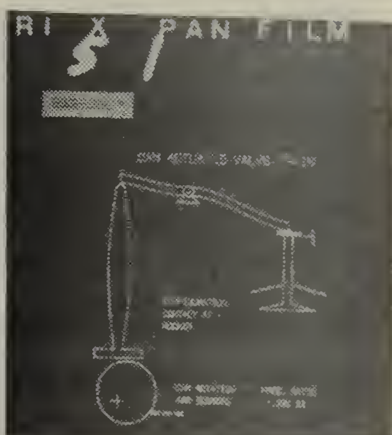


Photo 6.14 - 422 degrees

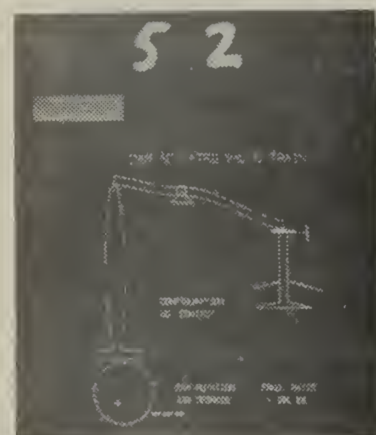


Photo 6.15 - 432 degrees

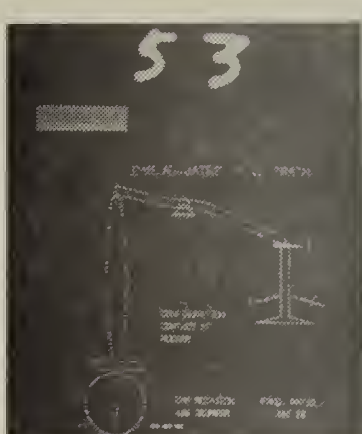


Photo 6.16 - 456 degrees

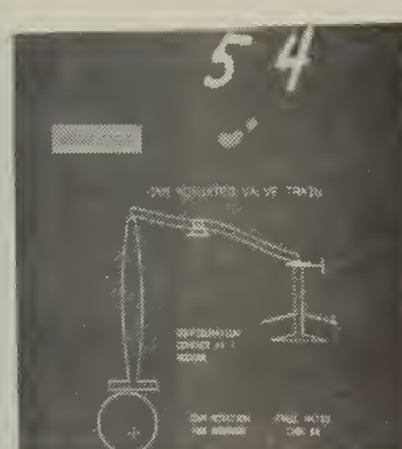


Photo 6.17 - 468 degrees

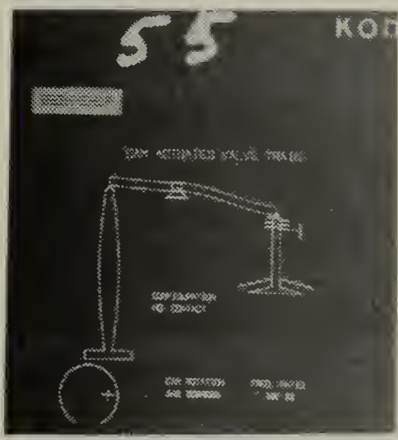


Photo 6.18 - 546 degrees

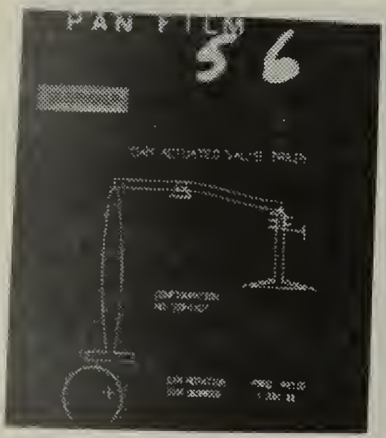


Photo 6.19 - 572 degrees

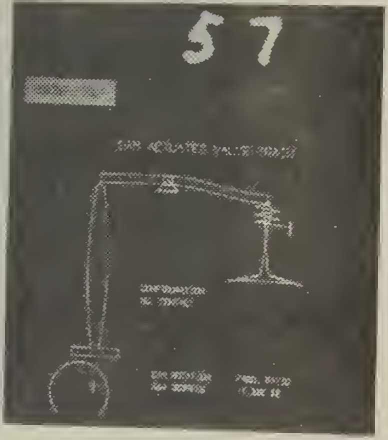


Photo 6.20 - 582 degrees

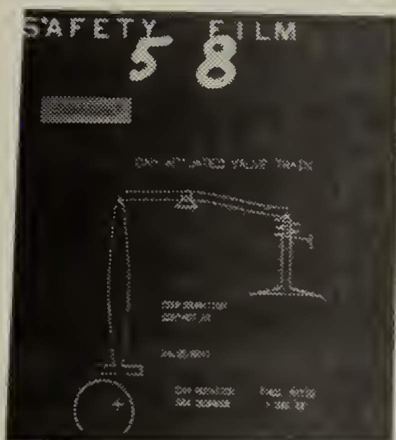


Photo 6.21 - 584 degrees

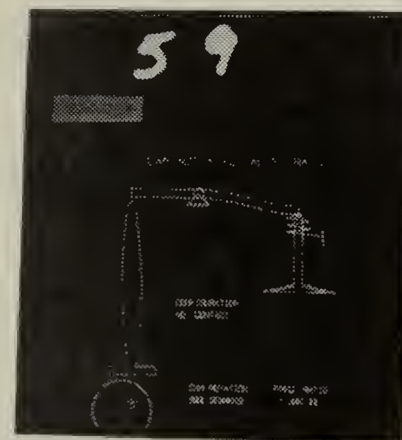


Photo 6.22 - 600 degrees

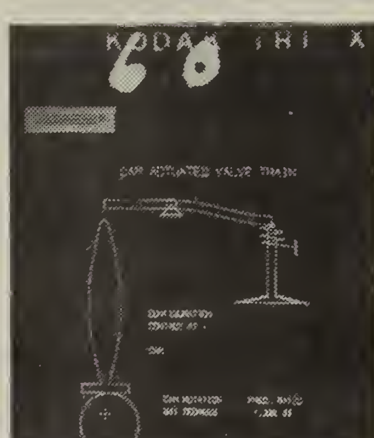


Photo 6.23 - 641 degrees

2, Example 1B, Three Clearances, 11,000 RPM, No Valve Seat

This analysis starts at 274 degrees of cam rotation, point (A). The following figures are utilized for the analysis.

a. Figure 6.35

(1) Displacement difference between cam and pushrod $(q_1 - q_9) \times 10^{-2}$

(2) Displacement of lower end of pushrod, i.e. follower $q_1 \times 10^{-1}$

(3) Cam profile, $q_9 \times 10^{-1}$

b. Figure 6.36

(1) Displacement difference between pushrod and rocker $(q_2 - q_8) \times 10^{-2}$

(2) Displacement difference between cam and pushrod $(q_1 - q_9) \times 10^{-2}$

c. Figure 6.37 Forces on the pushrod, $F_1 \times 10^3$

d. Figure 6.38 Forces on the rocker, $F_2 \times 10^2$

e. Figure 6.39 Displacement of Valve, $q_7 \times 10^2$

f. Figure 6.40 Axial oscillation of the pushrod, $(q_8 - q_1) \times 10^{-3}$

Figure 6.40 has been included for completeness and is not annotated.

The following reference points approximately correspond to the indicated degrees of cam rotation, Table 6.2

Reference Point	Degrees of Cam Rotation	Reference Point	Degrees of Cam Rotation
A	274	H	401
B	288	I	413
C	295	J	419
D	314	K	449
E	358	L	581
F	364	AA	634
G	382		

Table 6.2 Graph Reference Points and Degrees of Cam Rotation For Figures 6.35-6.39.

The analysis is similar to example 1A, except here the motion of the system is not impeded by the valve seat.

At point (A), the cam is at its lowest extreme and is starting to rise. The pushrod is floating between the cam and rocker, with downward motion.

At point (B), the cam and pushrod impact against one another and remain in contact for a short period, (B-C). The rising cam and force of impact create severe oscillations of the pushrod and a bounce occurs. The system remains intact, i.e. contact at the cam and rocker, (C-D). The valve has reached its highest extreme and reverses motion downwards.

At point (D), the pushrod and cam separate. Immediately, contact is also lost at the rocker and the pushrod floats free, (D-E).

At point (E), the rising pushrod now impacts against the rocker. This slows the motion of the pushrod sufficiently for the rising cam to again impact against the rocker, point (F). The system then remains in contact for a short period (F-G).

At point (G), the speed of the cam throws the pushrod clear. Immediately separation also occurs at the rocker - pushrod due to inertial effects. The pushrod floats free for a short time, (G-H), and then bounces at the rocker-pushrod clearance, point (I). The system next reaches its extreme value, point (K), i.e. the cam now with downward motion begins falling away from the pushrod. The valve spring, having reached the maximum extension, reverses the harmonic motion of the valve, upward. The rocker and pushrod are now in contact (J), and both have downward harmonic motion.

Finally, the compression of the valve spring slows the rocker valve assembly until the pushrod loses contact at the rocker and again floats free, point (L).

At point (AA), 634 degrees of cam rotation, the cycle then repeats.

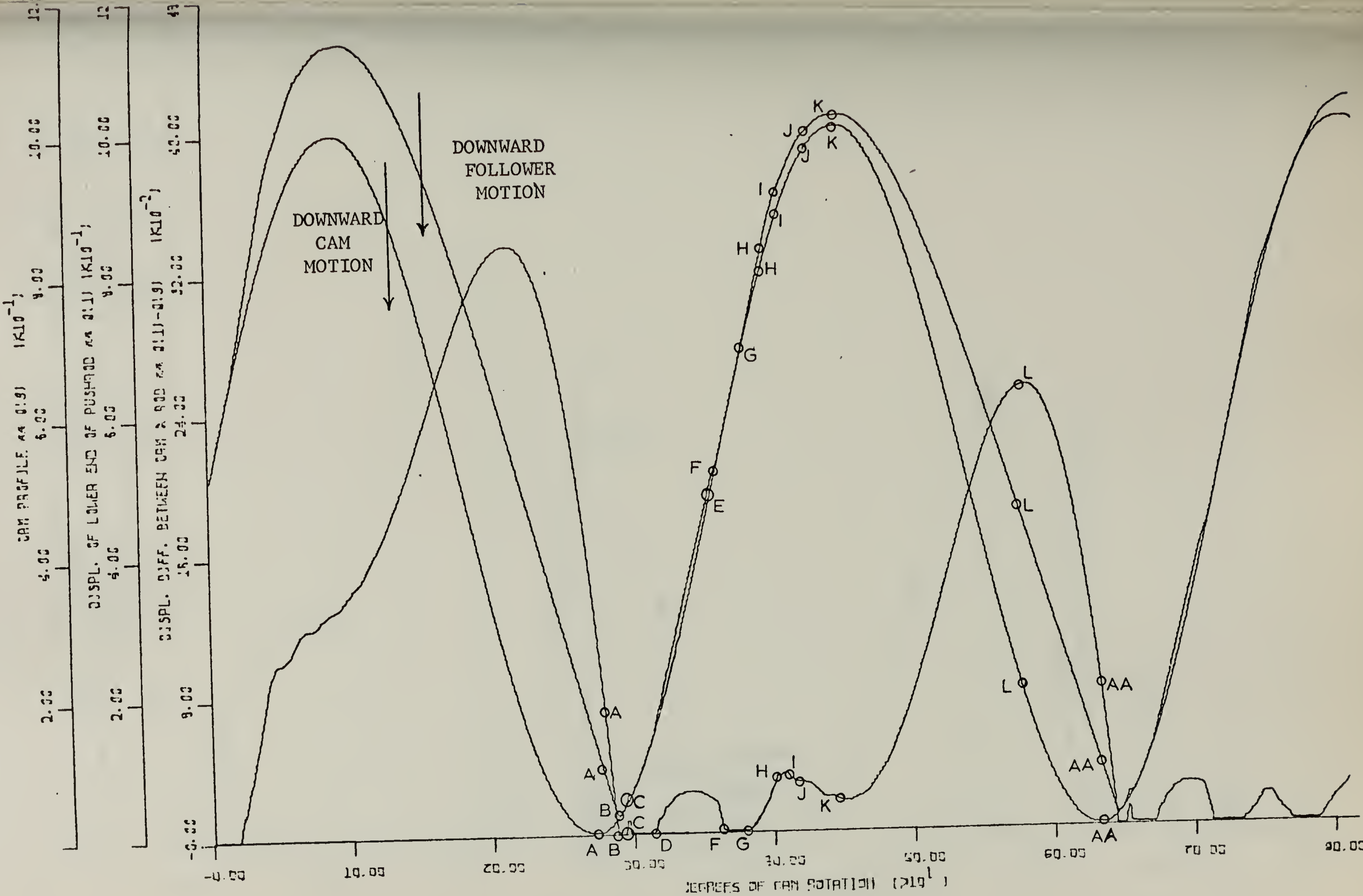


Figure 6.35 THREE CLEARANCES 11,000 RPM NO VALVE SEAT

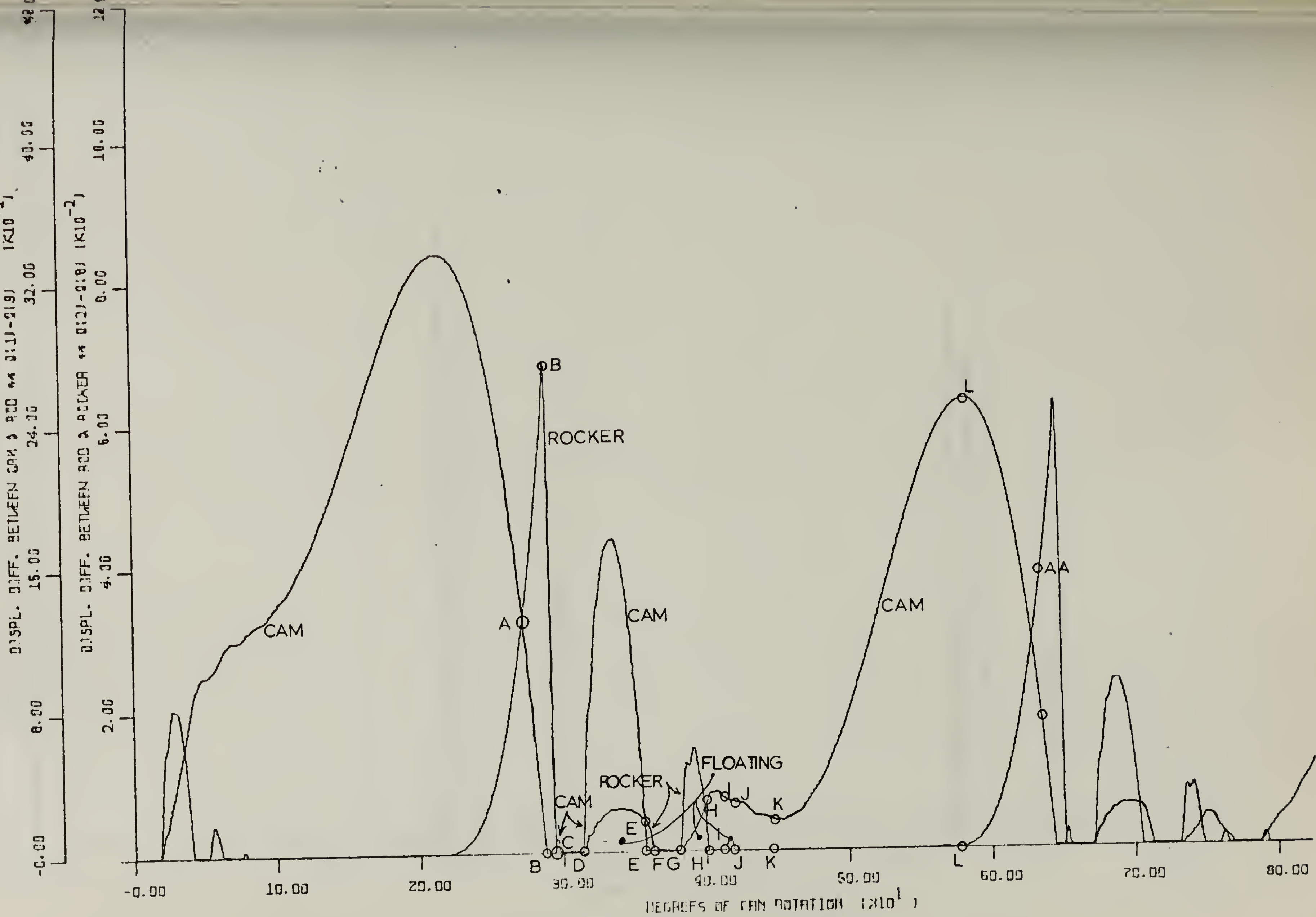


Figure 6.36 THREE CLEARANCES 11,000 RPM NO VALVE SEAT

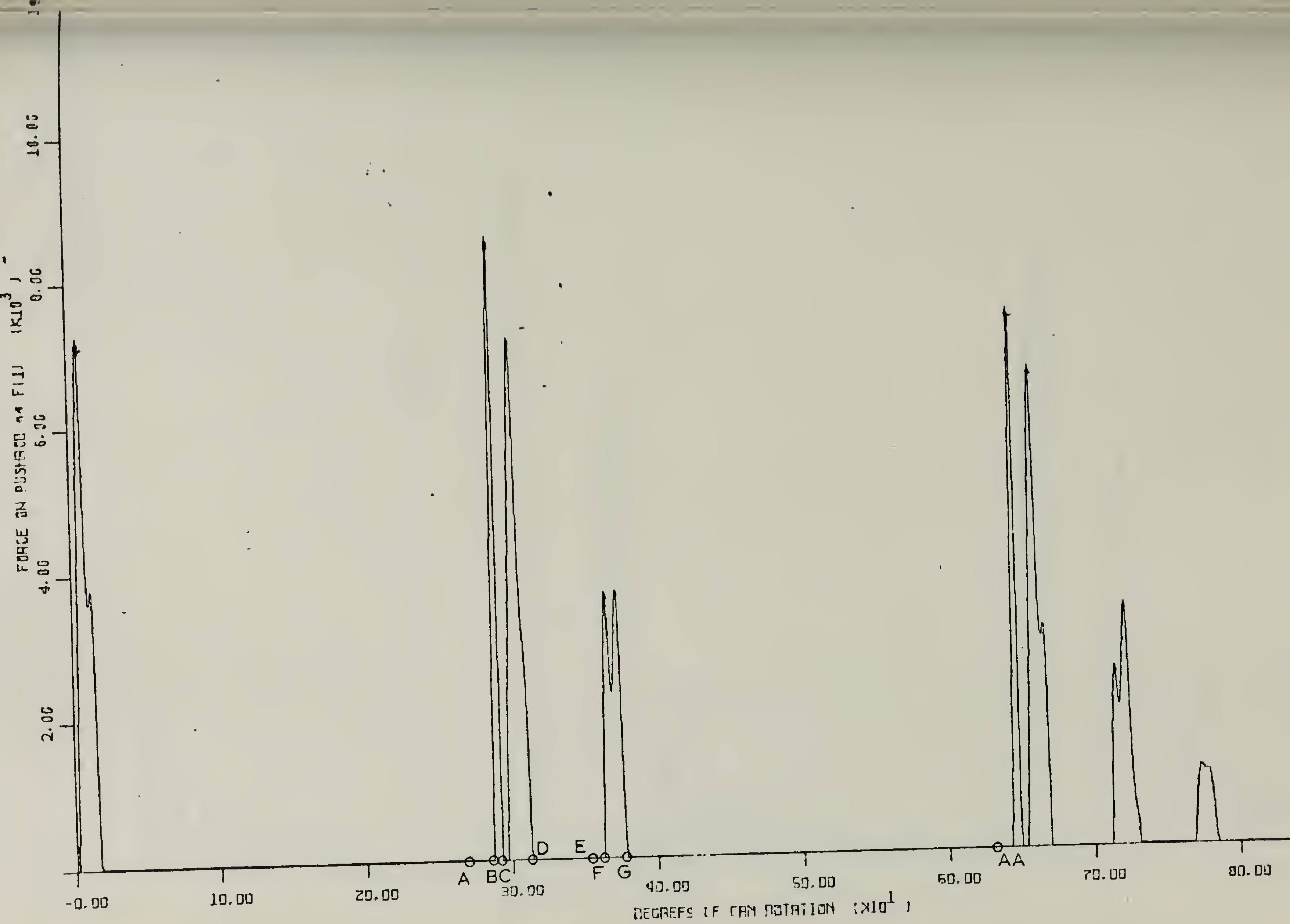


Figure 6.37 THREE CLEARANCES 11,000 RPM NO VALVE SEAT

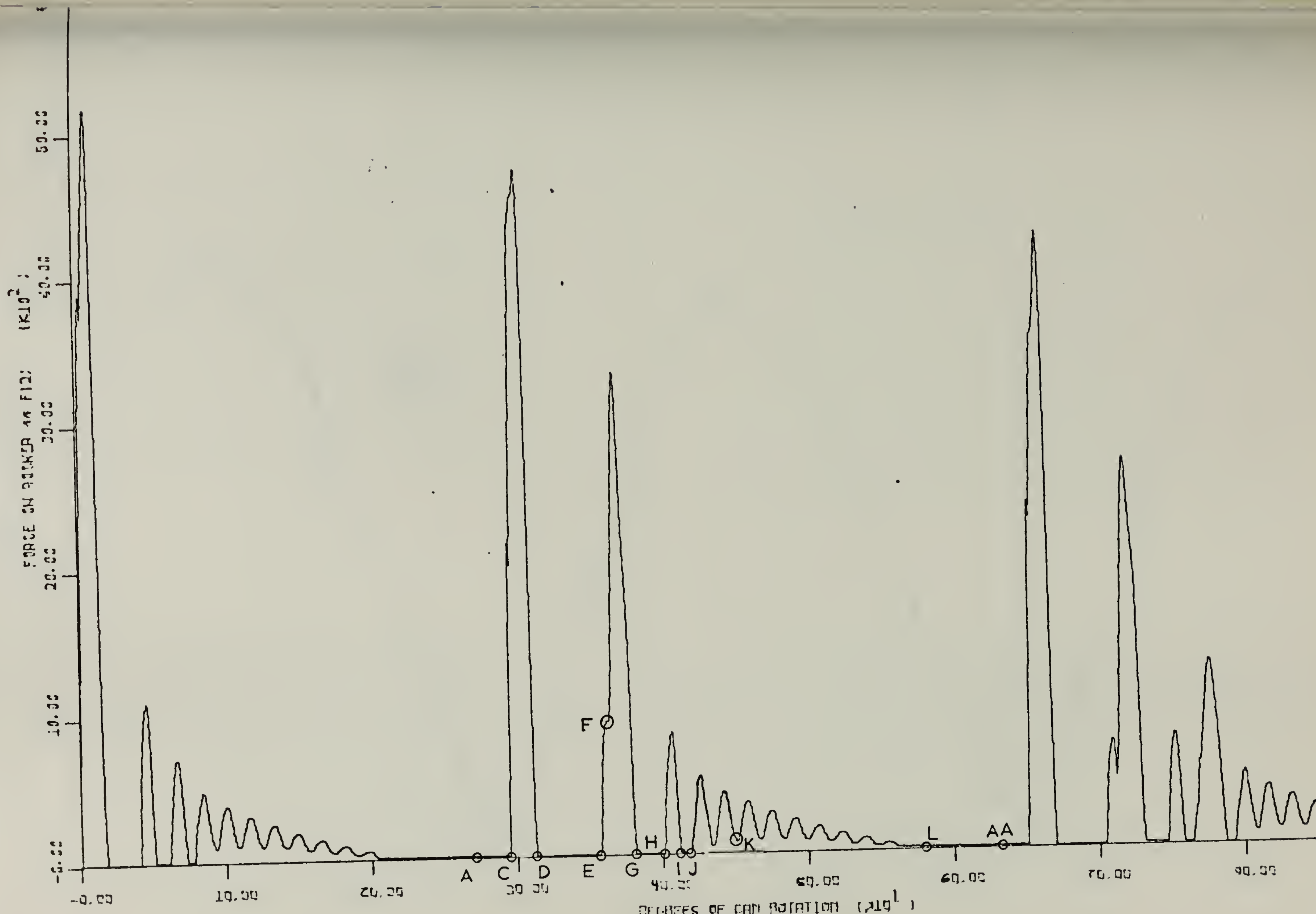


Figure 6.38 THREE CLEARANCES 11,000 RPM NØ VALVE SEAT

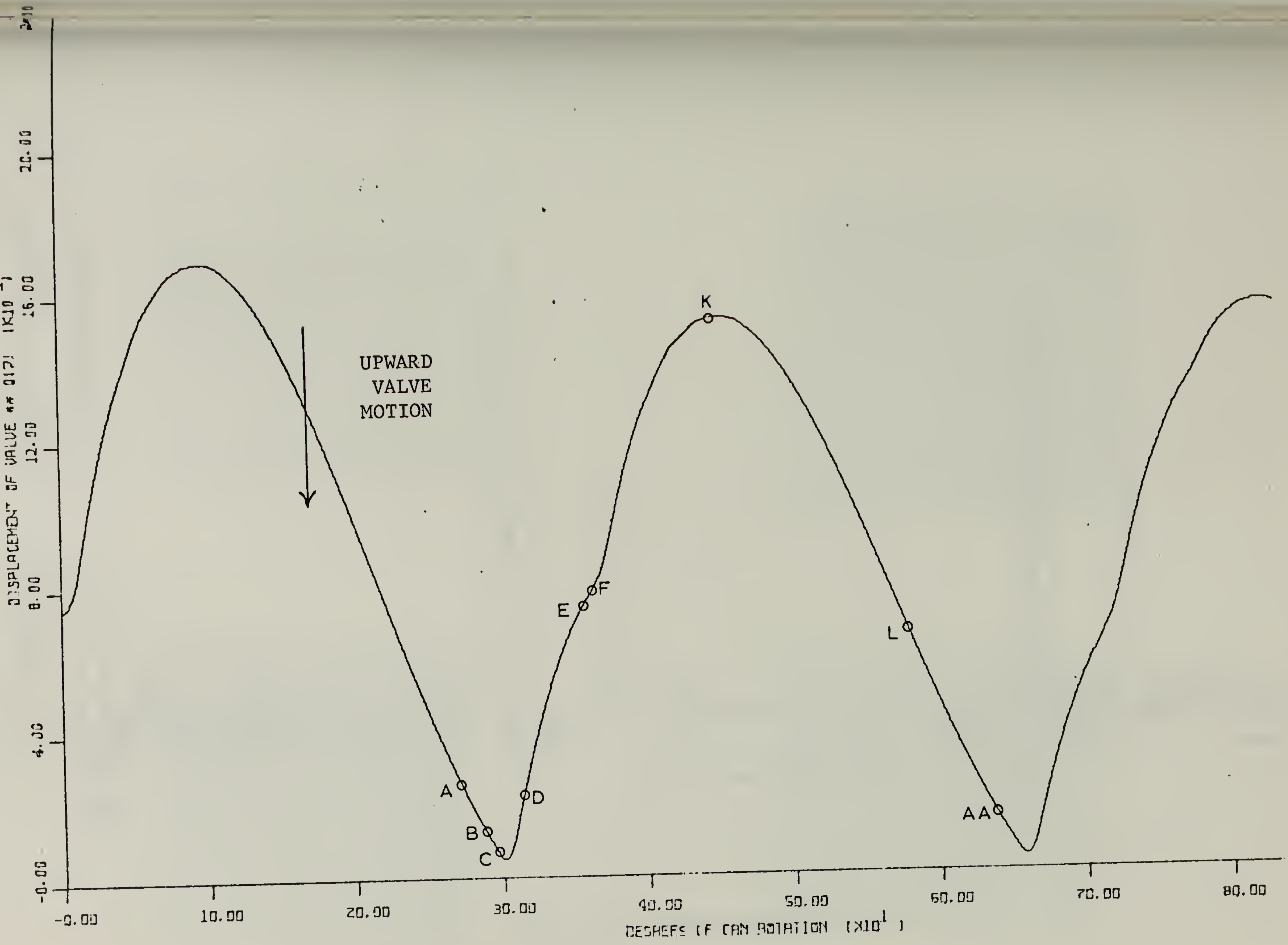


Figure 6.39 THREE CLEARANCES 11,000 RPM NO VALVE SEAT

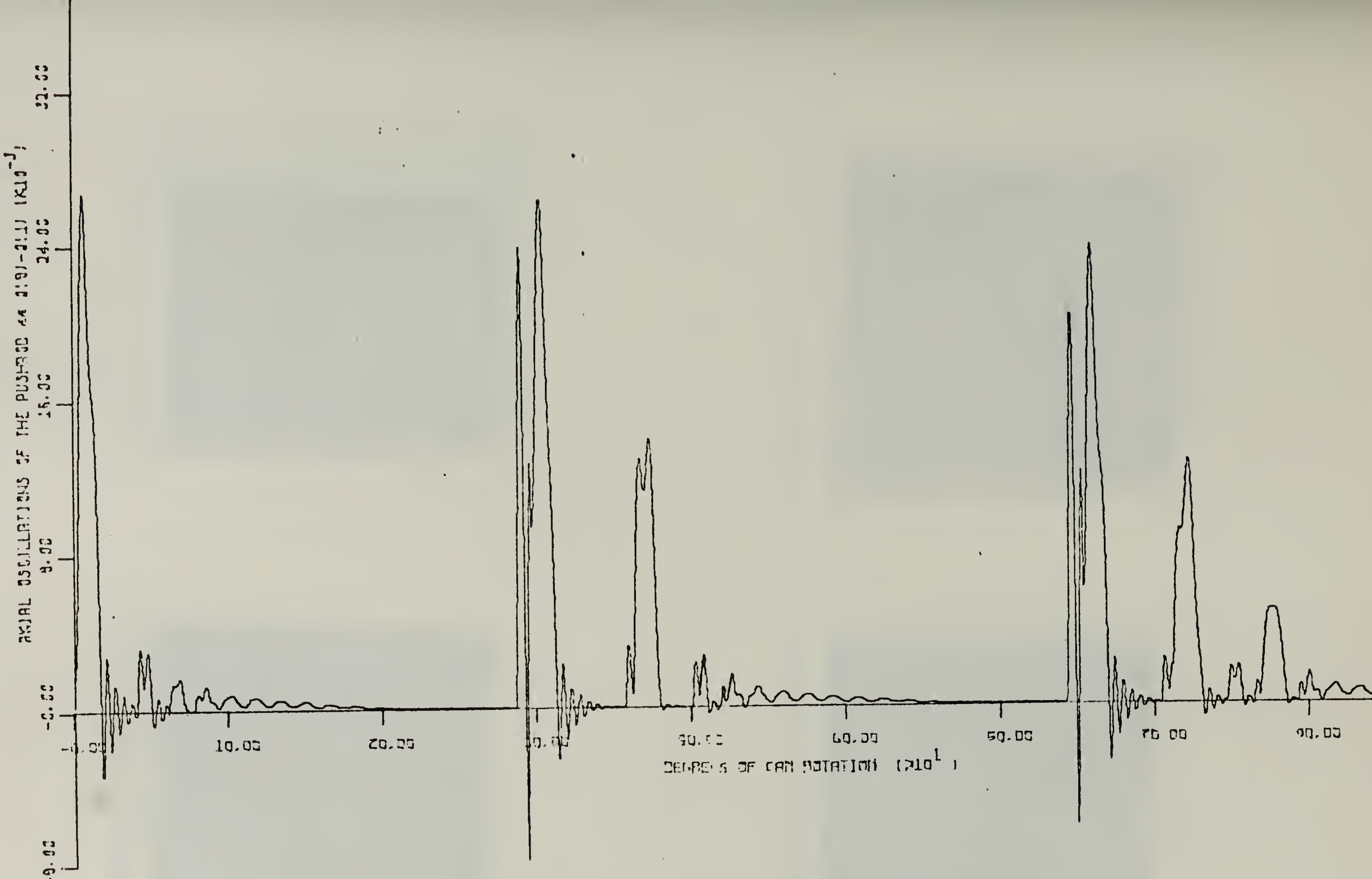


Figure 6.40 THREE CLEARANCES

11,000 RPM NO VALVE SEAT

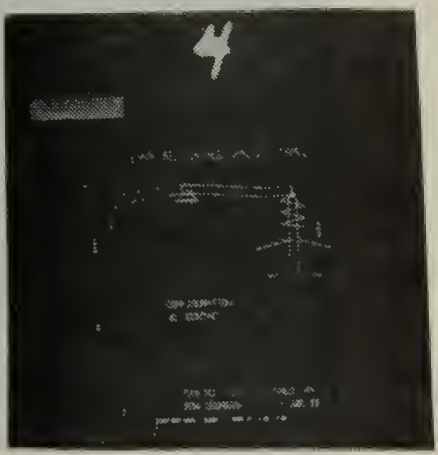


Photo 6.24 - 634 degrees

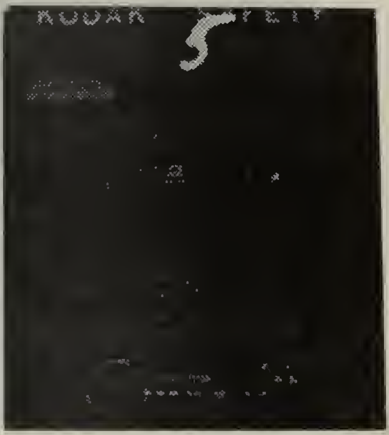


Photo 6.25 - 646 degrees

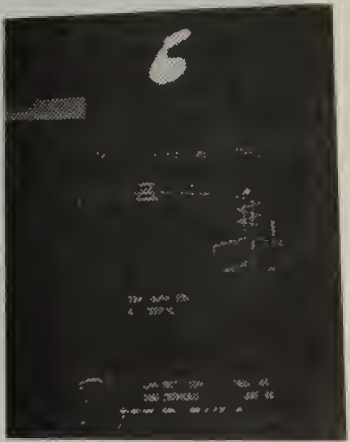


Photo 6.26 - 652 degrees

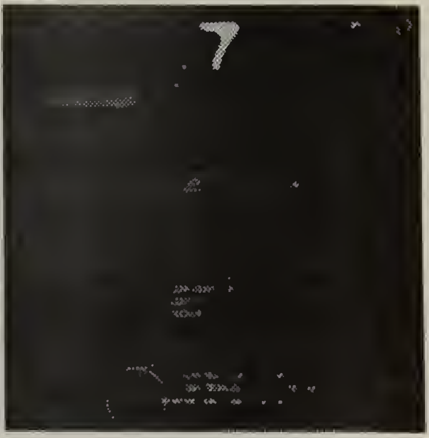


Photo 6.27 - 654 degrees

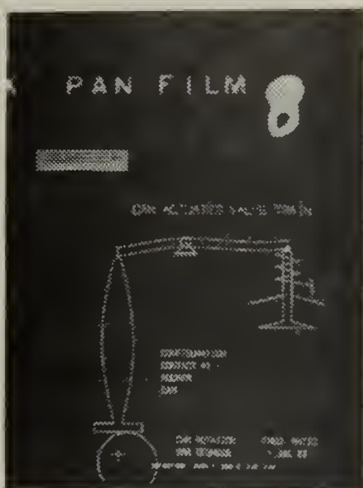


Photo 6.28 - 658 degrees

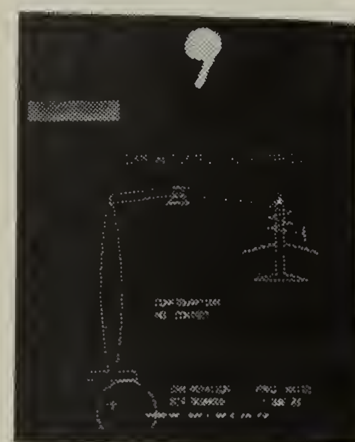


Photo 6.29 - 674 degrees

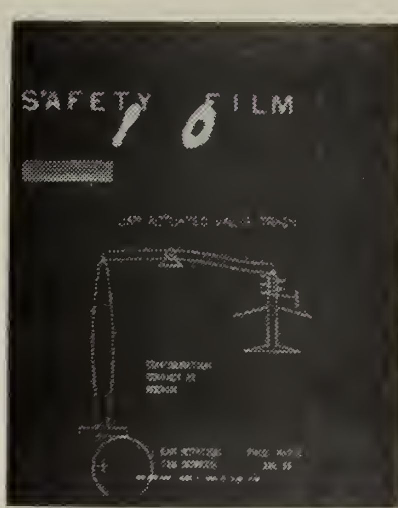


Photo 6.30 - 708 degrees

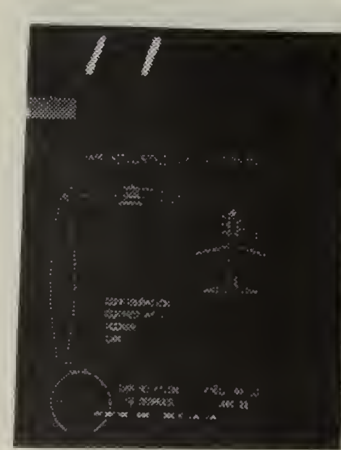


Photo 6.31 - 716 degrees

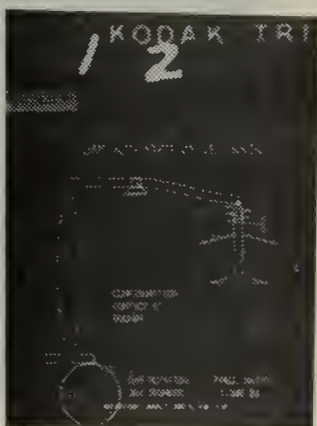


Photo 6.32 - 364 degrees

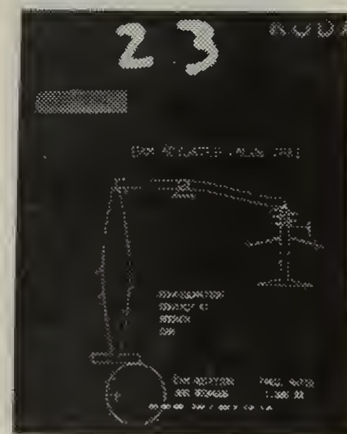


Photo 6.33 - 366 degrees

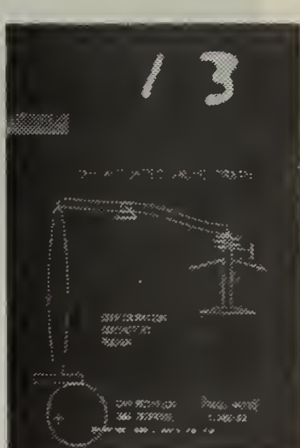


Photo 6.34 - 382 degrees

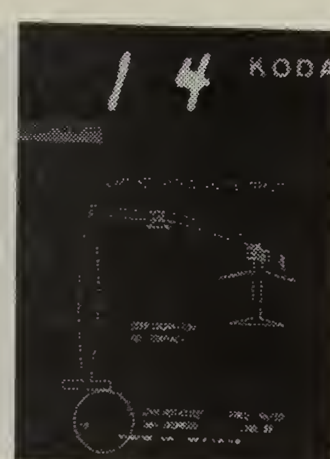


Photo 6.35 - 384 degrees

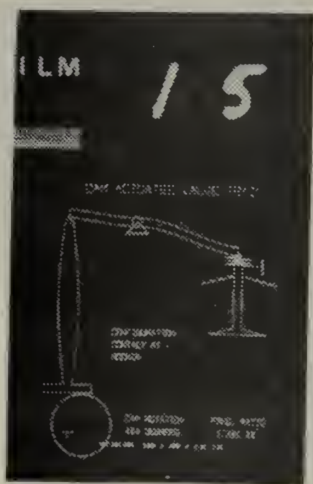


Photo 6.36 - 404 degrees

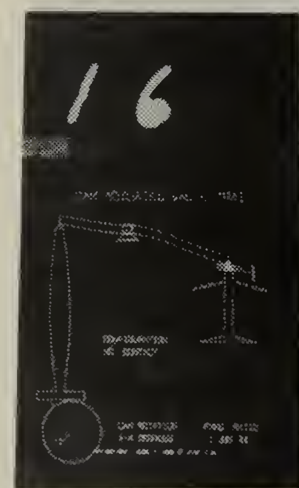


Photo 6.37 - 414 degrees

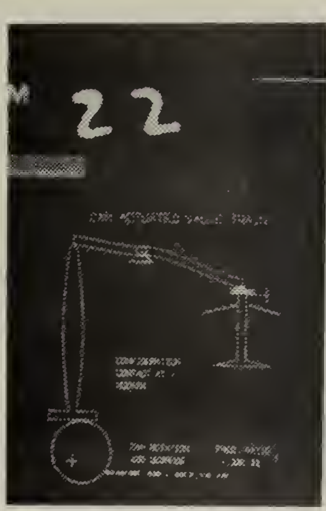


Photo 6.38 - 422 degrees

Photo 6.39 - 450 degrees

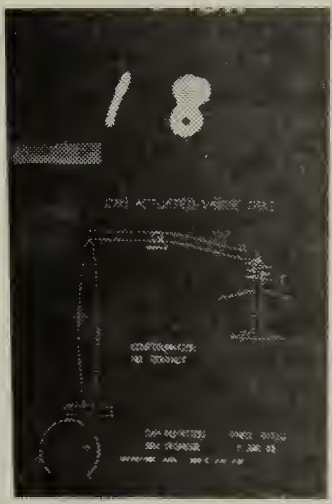


Photo 6.40 - 464 degrees

Photo 6.41 - 572 degrees

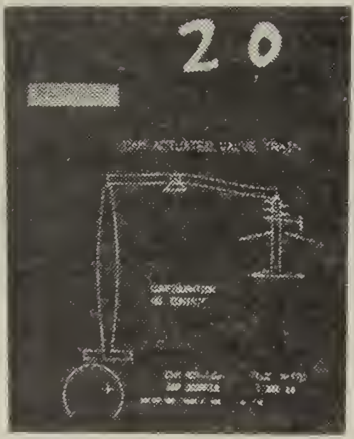


Photo 6.42 - 582 degrees

3. <u>Example 1C, Two Clearances, 11,000 RPM, Valve Seat</u>

This analysis will commence at 333 degrees of cam rotation. One cycle will be discussed, from point (A) to point (AA). The following figures are utilized for the analysis:

a. Figure 6.41 Cam profile and displacement of pushrod, $q_9 \times 10^{-1}$ and $q_1 \times 10^{-1}$

b. Figure 6.42

   (1) Cam profile, $q_9 \times 10^{-1}$

   (2) Displacement difference between cam and pushrod, $(q_1 - q_9) \times 10^{-2}$

c. Figure 6.43 Force on pushrod, $F_1 \times 10^3$

d. Figure 6.44 Force on the valve, $F_7 \times 10^3$

e. Figure 6.45 Displacement of valve and valve seat, $q_7 \times 10^{-1}$ and $q_{10} \times 10^{-1}$

f. Figure 6.46 Axial oscillation of the pushrod, $(q_8 - q_1) \times 10^{-2}$

g. Figure 6.47 Axial oscillation of the valve stem, $(q_7 - q_6) \times 10^{-3}$

Figures 6.46 and 6.47 have been included for completeness and are not annotated.

The following reference points approximately correspond to the indicated degrees of cam rotation, Table 6.3.

Reference Point	Degrees of Cam Rotation
A	333
B	340
C	379
D	398
E	448
F	467
G	483

Reference Point	Degrees of Cam Rotation
H	600
I	606
J	612
K	681
L	688
AA	693

Table 6.3 Graph Reference Points and Degrees of Cam Rotation For Figures 6.41 - 6.45

The analysis is similar to that for examples 1A and 1B. For this example the pushrod is pinned to the rocker and thus moves as a part of the system rather than floating free.

At point (A), impact against the pushrod by the cam has just occurred. This has created large system oscillations, first in the pushrod and later in the valve stem. The cam and pushrod are rising. At points (B) and (D) the pushrod is thrown clear of the cam due to its high speed of rotation. At point (E) the cam has reached its highest position and starts to fall. The valve has reached its lowest position and starts its upward travel.

At point (F) the pushrod and rocker come into contact with very little force and travel together for a short distance. Separation again occurs at point (G) when the cam falls away from the pushrod. The valve in its upward travel impacts against the valve seat at point (H). This retards the motion of the pushrod just before contact

at the cam was due to occur (I). The large oscillations created by the valve seating cause the valve to bounce and leave contact at the seat, point (J).

Meanwhile the cam has reached its lowest extreme and begins its upward travel. At point (K) the rising cam impacts against the oscillating pushrod and forces the system to return to its harmonic motion. The cycle then repeats, point (AA).

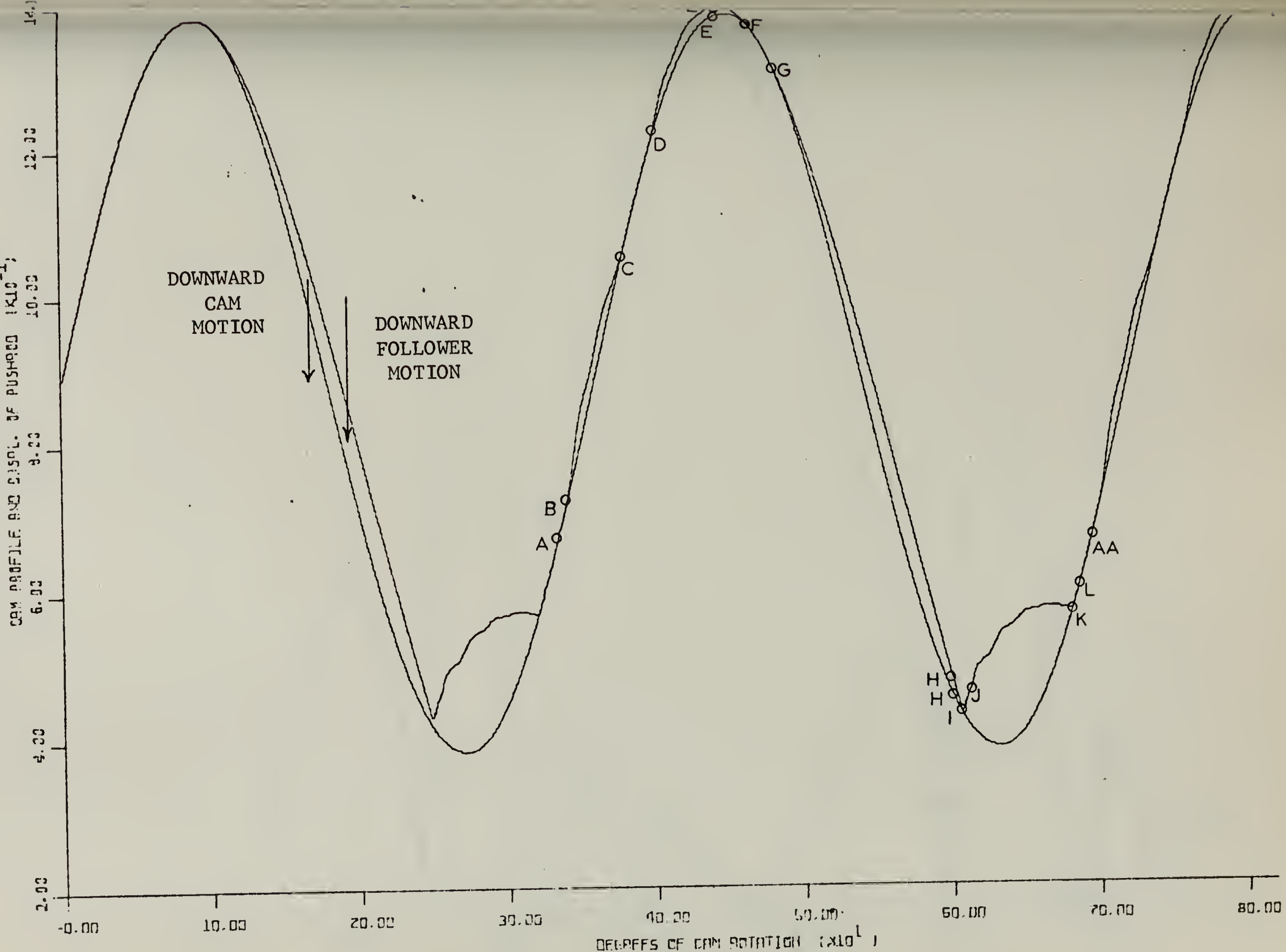


Figure 6.41 TWO CLEARANCES 11,000 RPM VALUE SEAT

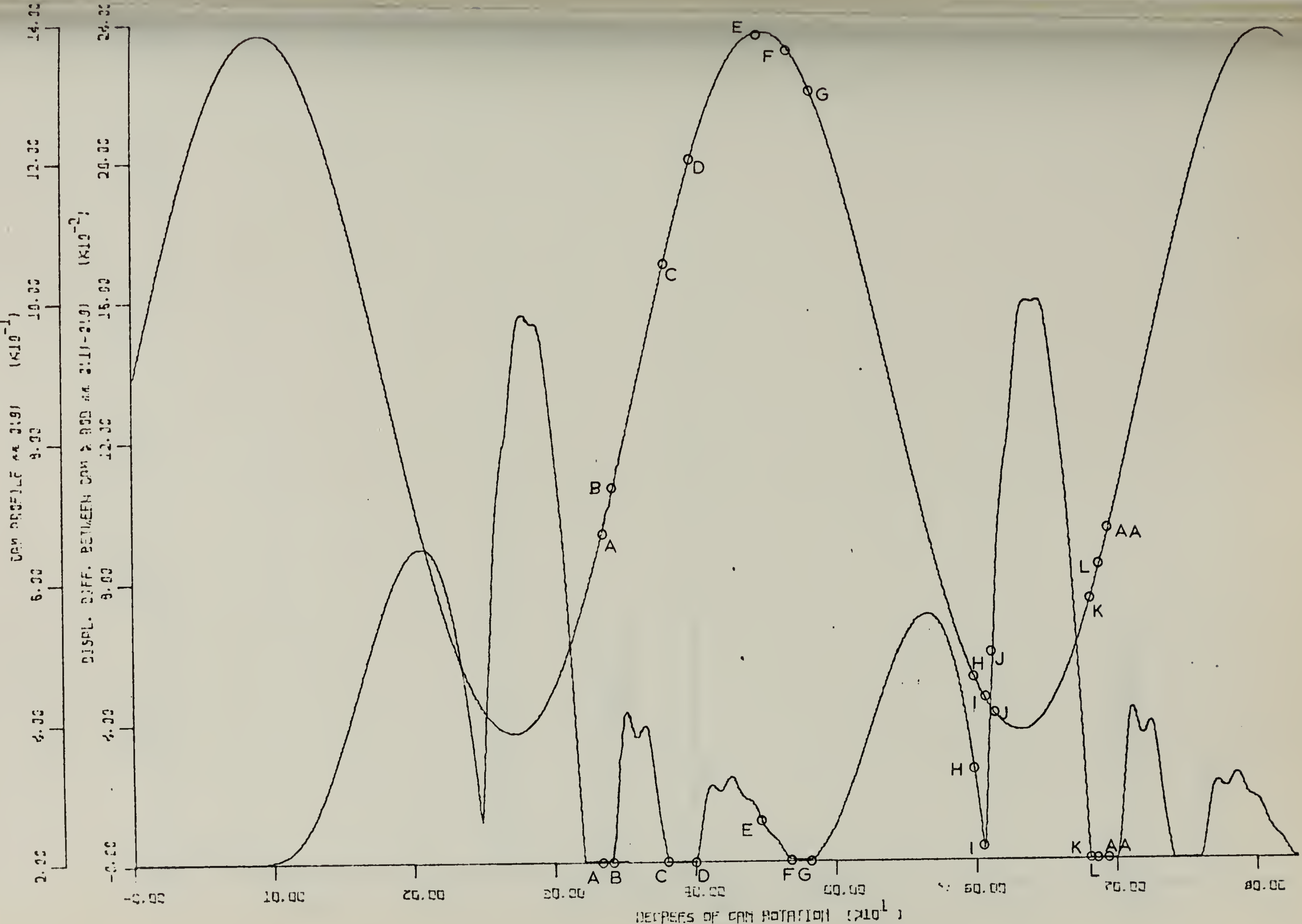


Figure 6.42 TWO CLEARANCES 11,000 RPM VALVE SEAT

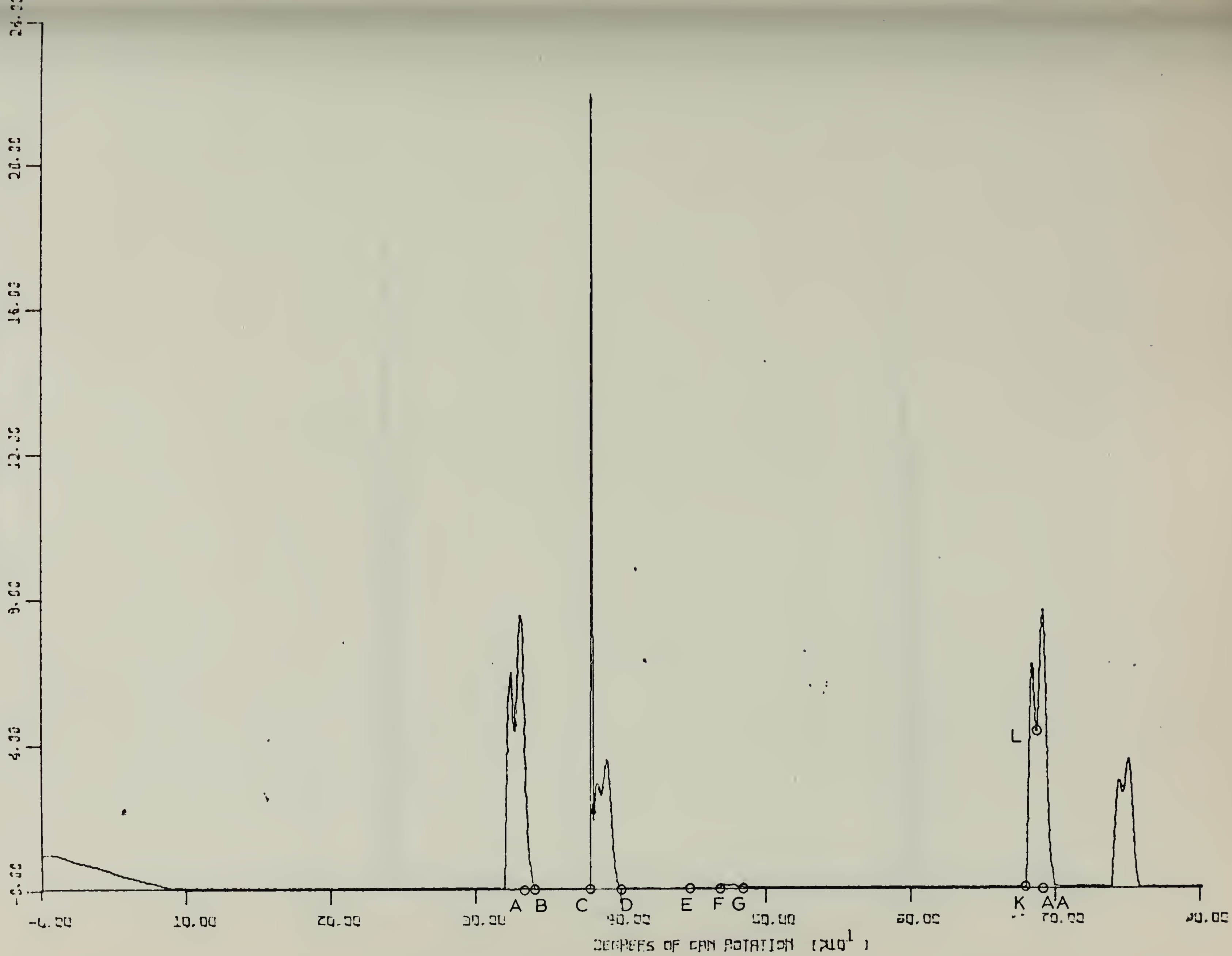


Figure 6.43 TWO CLEARANCES 11,000 RPM VALVE SEAT

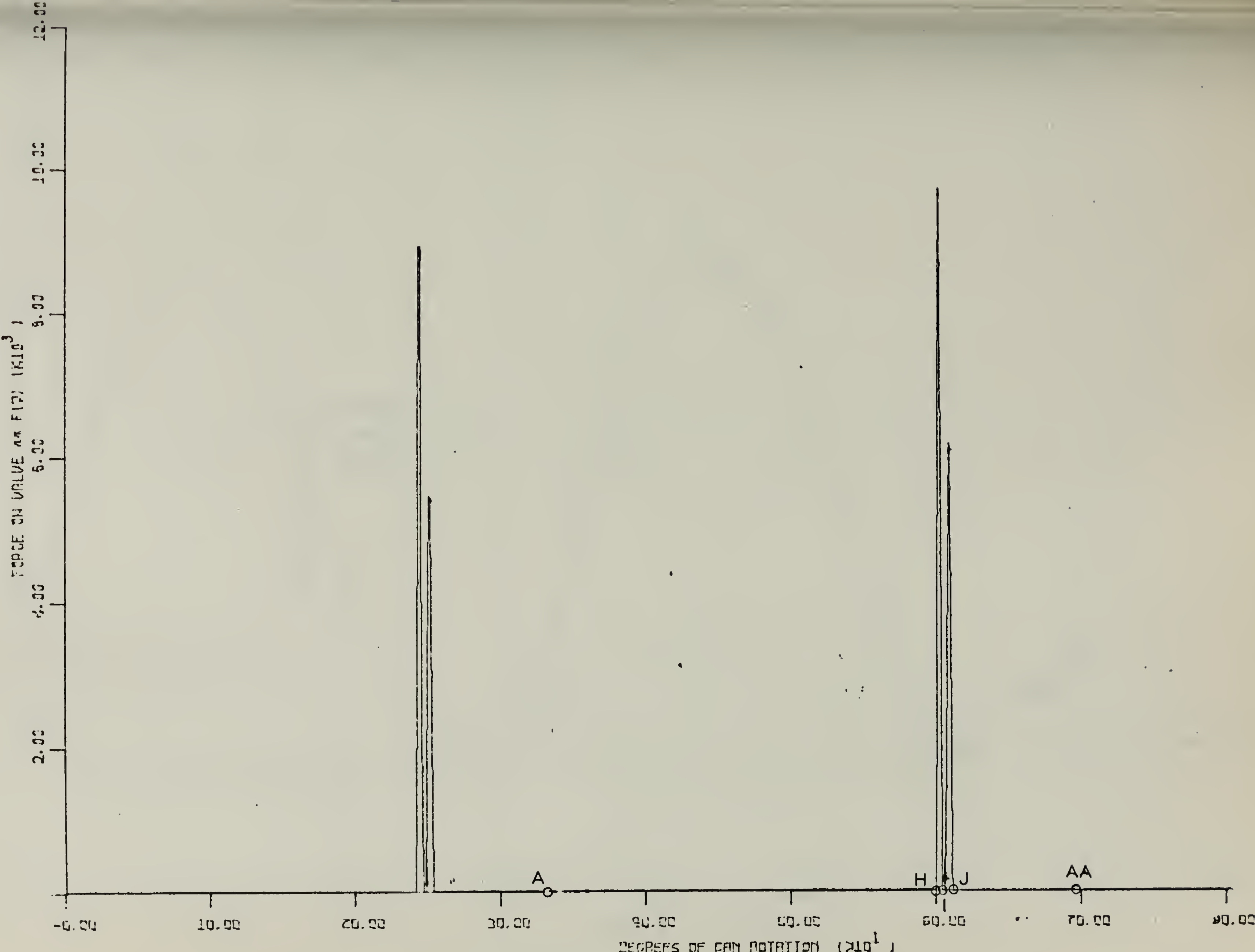


Figure 6.44 TWO CLEARANCES 11,000 RPM VALVE SEAT

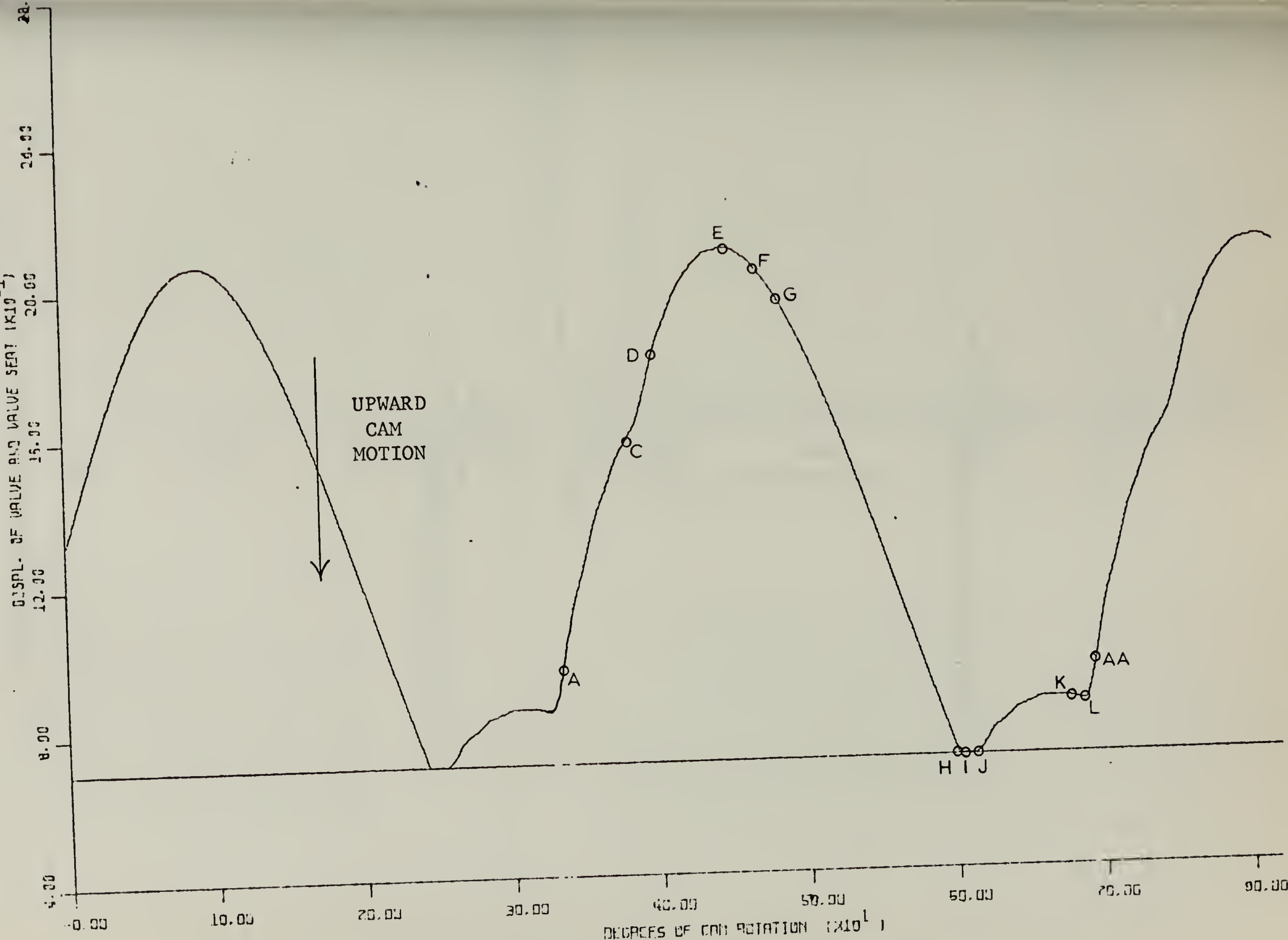


Figure 6.45 TWO CLEARANCES 11.000 RPM VALVE SEAT

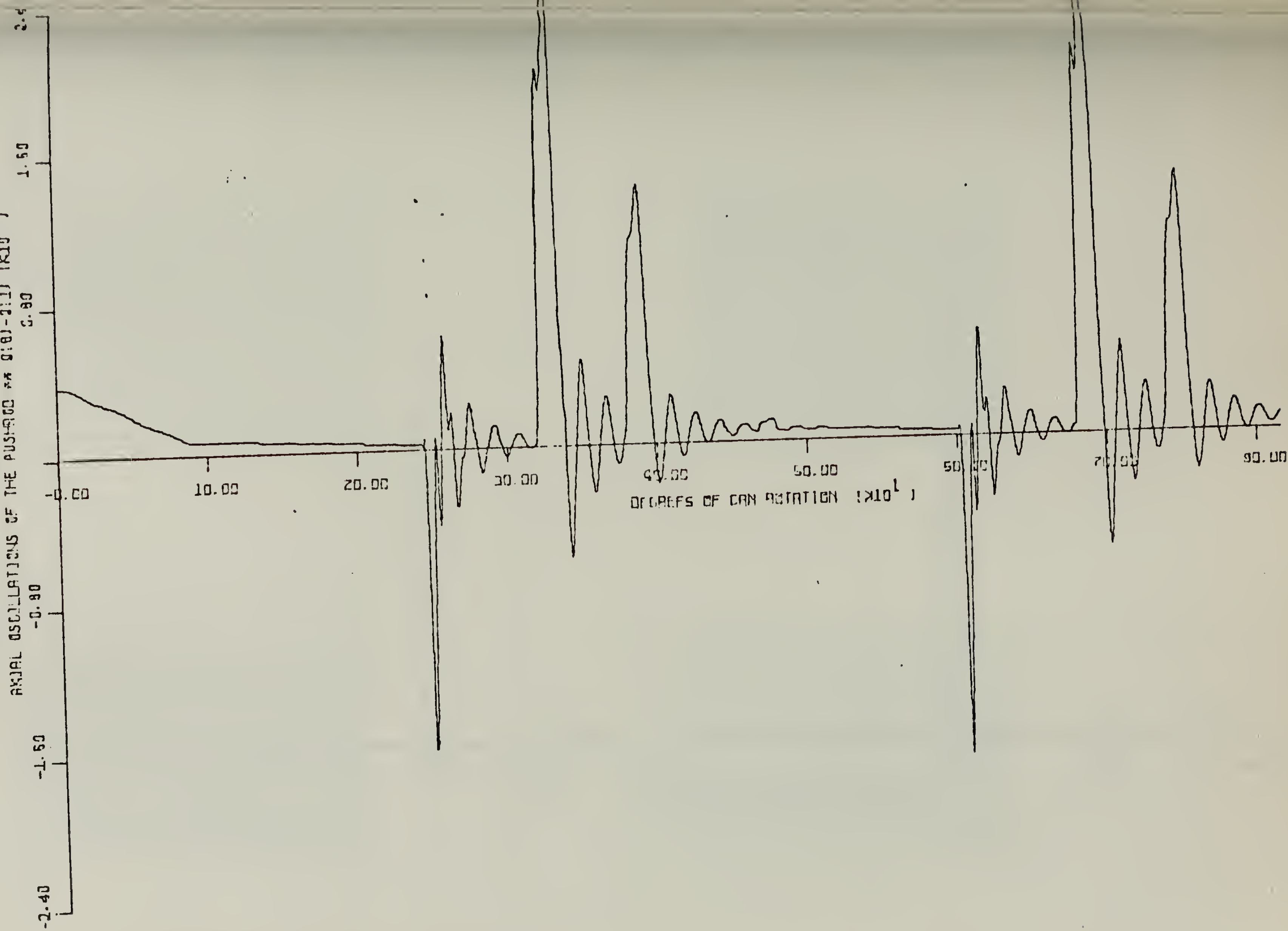


Figure 6.46 TWO CLEARANCES 11,000 RPM VALVE SEAT

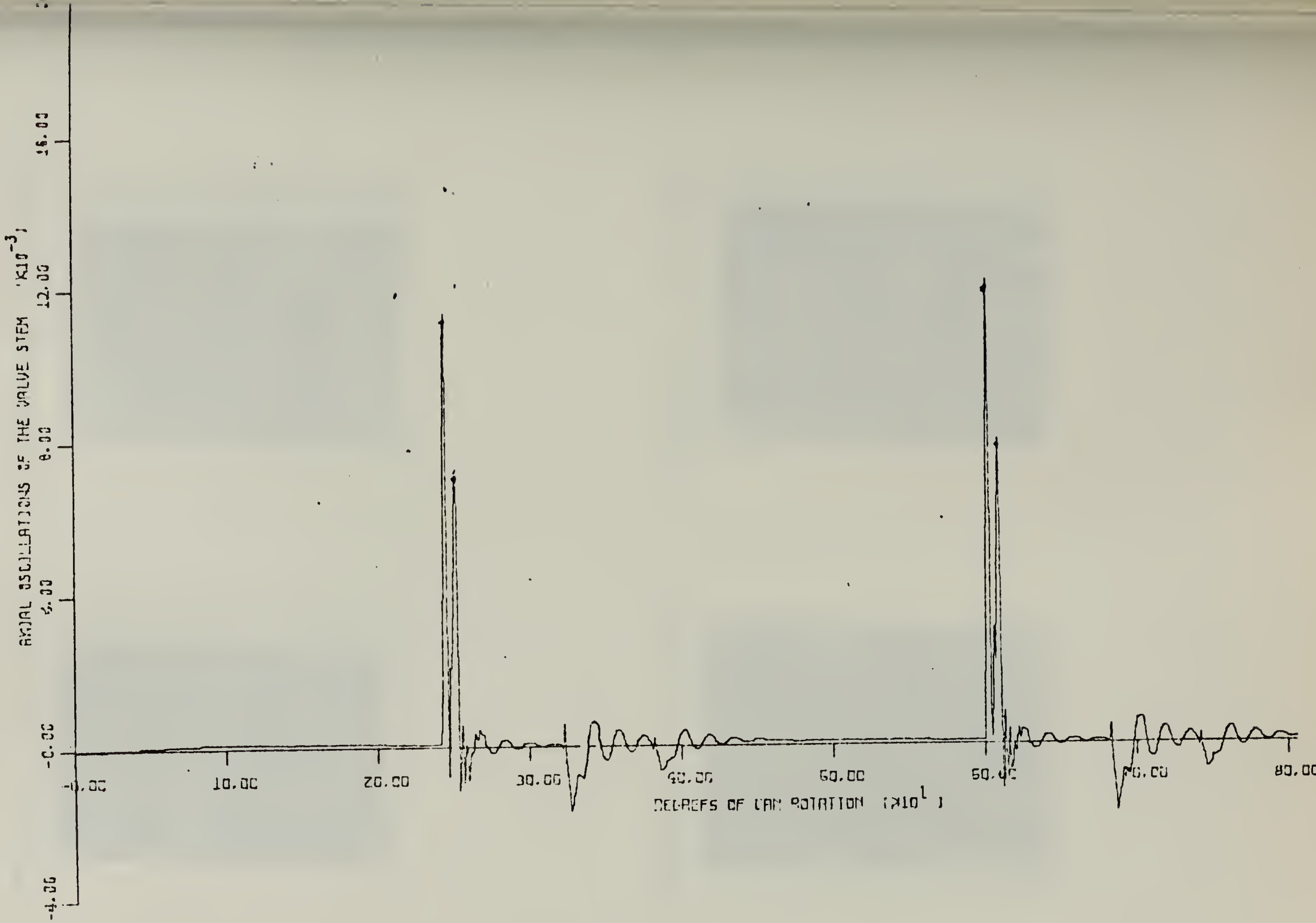


Figure 6.47 TWO CLEARANCES 11,000 RPM VALVE SEAT

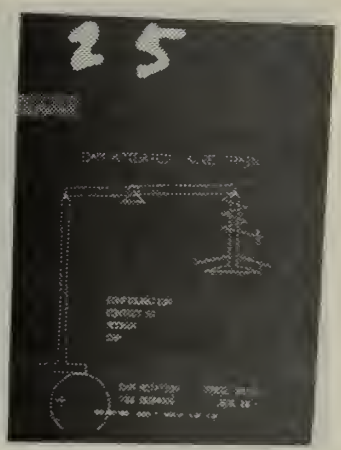


Photo 6.43 - 700 degrees

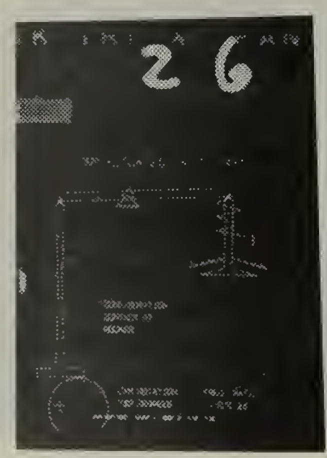


Photo 6.44 - 702 degrees

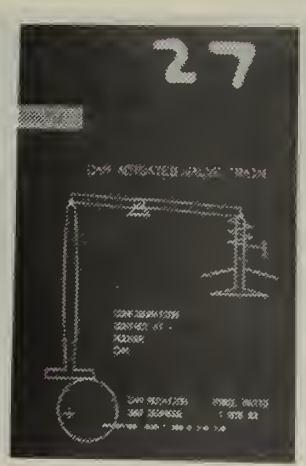


Photo 6.45 - 382 degrees

Photo 6.46 - 400 degrees

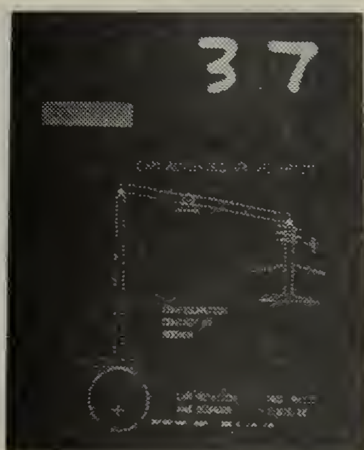


Photo 6.47 - 448 degrees

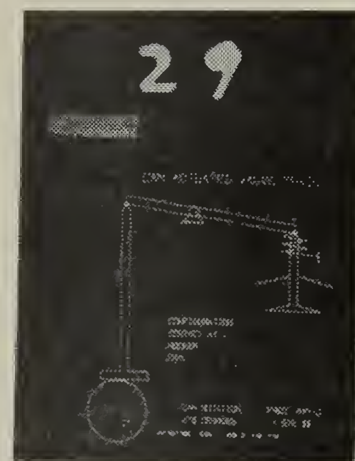


Photo 6.48 - 470 degrees

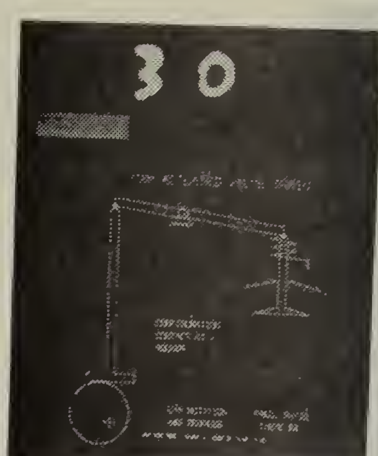


Photo 6.49 - 482 degrees

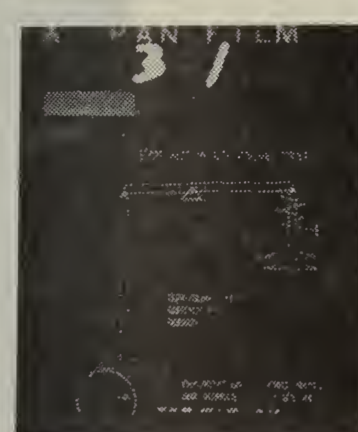


Photo 6.50 - 560 degrees

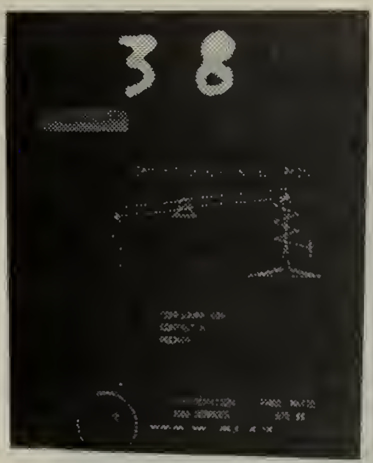


Photo 6.51 - 602 degrees

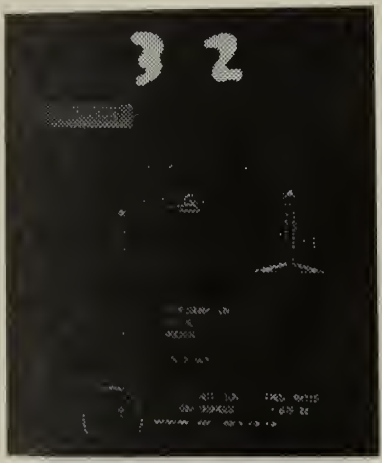


Photo 6.52 - 604 degrees

Photo 6.53 - 606 degrees

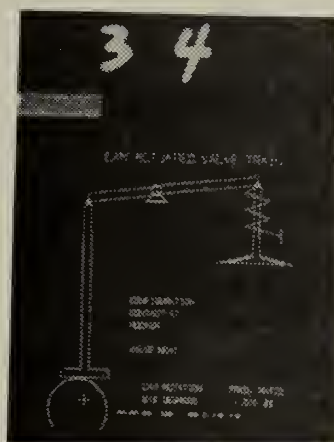


Photo 6.54 - 610 degrees

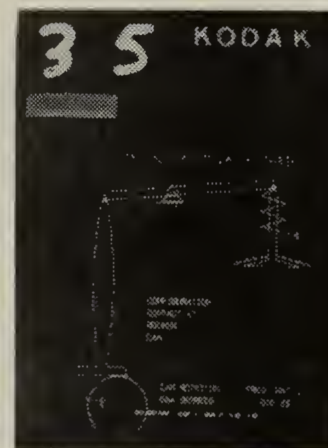


Photo 6.55 - 684 degrees

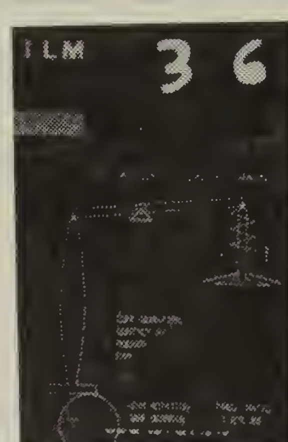


Photo 6.56 - 690 degrees



4. <u>Example 1D, Two Clearances, 11,000 RPM, No Valve Seat</u>

This analysis will commence at 274 degrees of cam rotation, point (A). The following figures are utilized for the analysis.

a. Figure 6.48

(1) Cam profile, $q_9 \times 10^{-1}$

(2) Displacement of lower end of pushrod, $q_1 \times 10^{-1}$

(3) Displacement difference between cam and pushrod, $(q_1 - q_9) \times 10^{-2}$

b. Figure 6.49, Force on pushrod, $F_1 \times 10^3$

c. Figure 6.50, Displacement of valve, $q_7 \times 10^{-1}$

d. Figure 6.51, Axial oscillation of the pushrod, $(q_9 - qq_1) \times 10^{-2})$

Figure 6.51 has been included for completeness and is not annotated.

The following reference points approximately correspond to the indicated degrees of cam rotation, Table 6.4.

Reference Point	Degrees of Cam Rotation
A	274
B	295
C	314
D	376

Reference Point	Degrees of Cam Rotation
E	398
F	448
AA	634

Table 6.4 Graph Reference Points and Degrees of Cam Rotation For Figures 6.48 - 6.50

The analysis is similar to the above examples. The valve seat position is higher than the maximum excursion of the valve, thus contact does not occur between the seat and valve. The remaining points of interest are then only associated with clearance 1, between pushrod and cam.

At point (A) the cam has reached its lowest extreme and begins rising. The pushrod continues its downward harmonic motion, the valve its harmonic motion upward. At point (B), the rising cam impacts against the pushrod. The motion of the pushrod is reversed. The severe oscillations and inertial effects first insure positive contact and then cause the pushrod and cam to separate, point (C). As the upward motion of the pushrod slows, the rising cam again contacts the pushrod, point (D). Since the motion of these two members is similar, the impact force is not as severe.

At point (E), the speed of cam rotation throws the pushrod off. Harmonic motion of the system then results. The cycle repeats at point (AA).

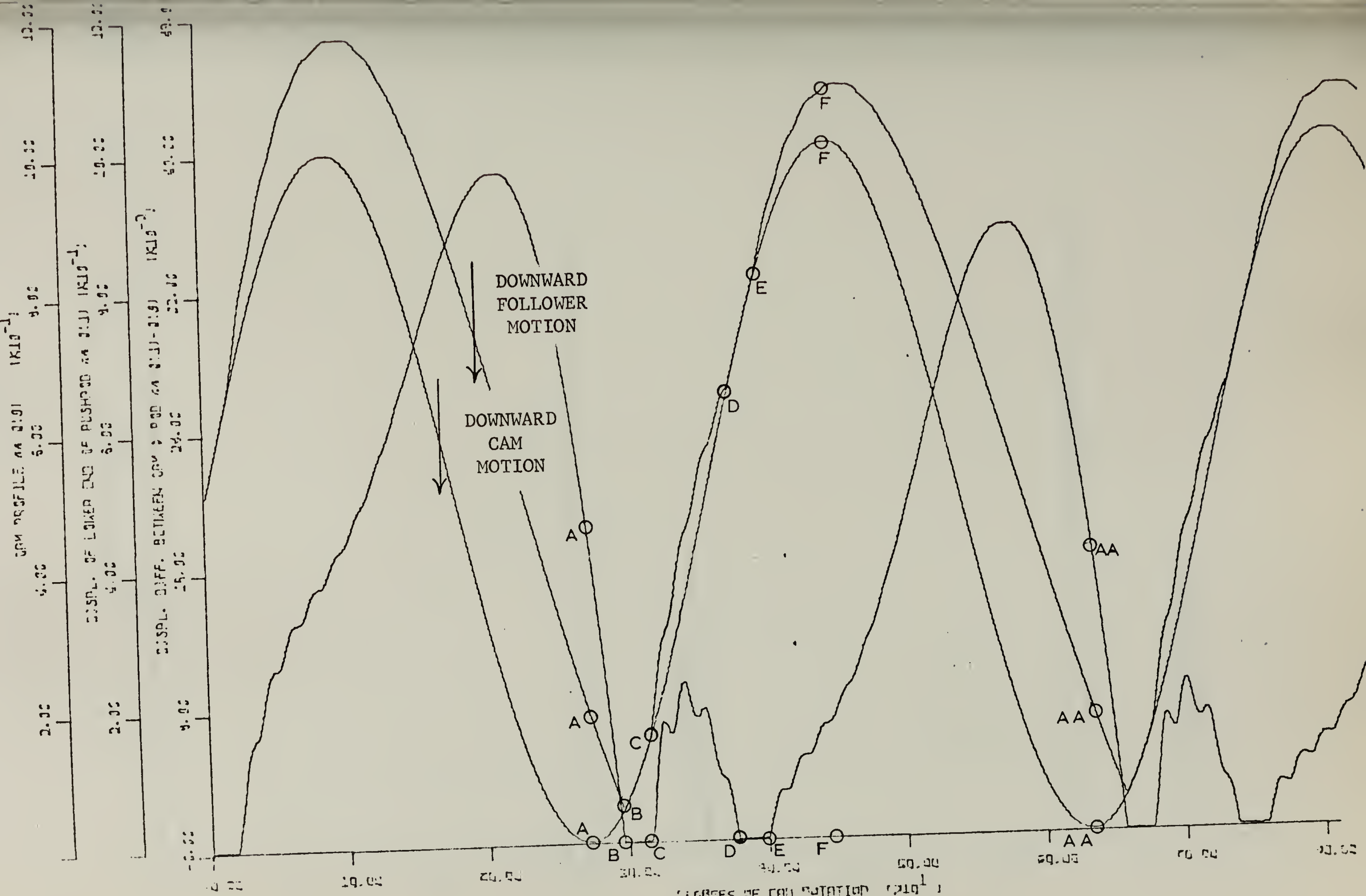


Figure 6.48 TWO CLEARANCES 11,000 RPM NO VALVE SEAT

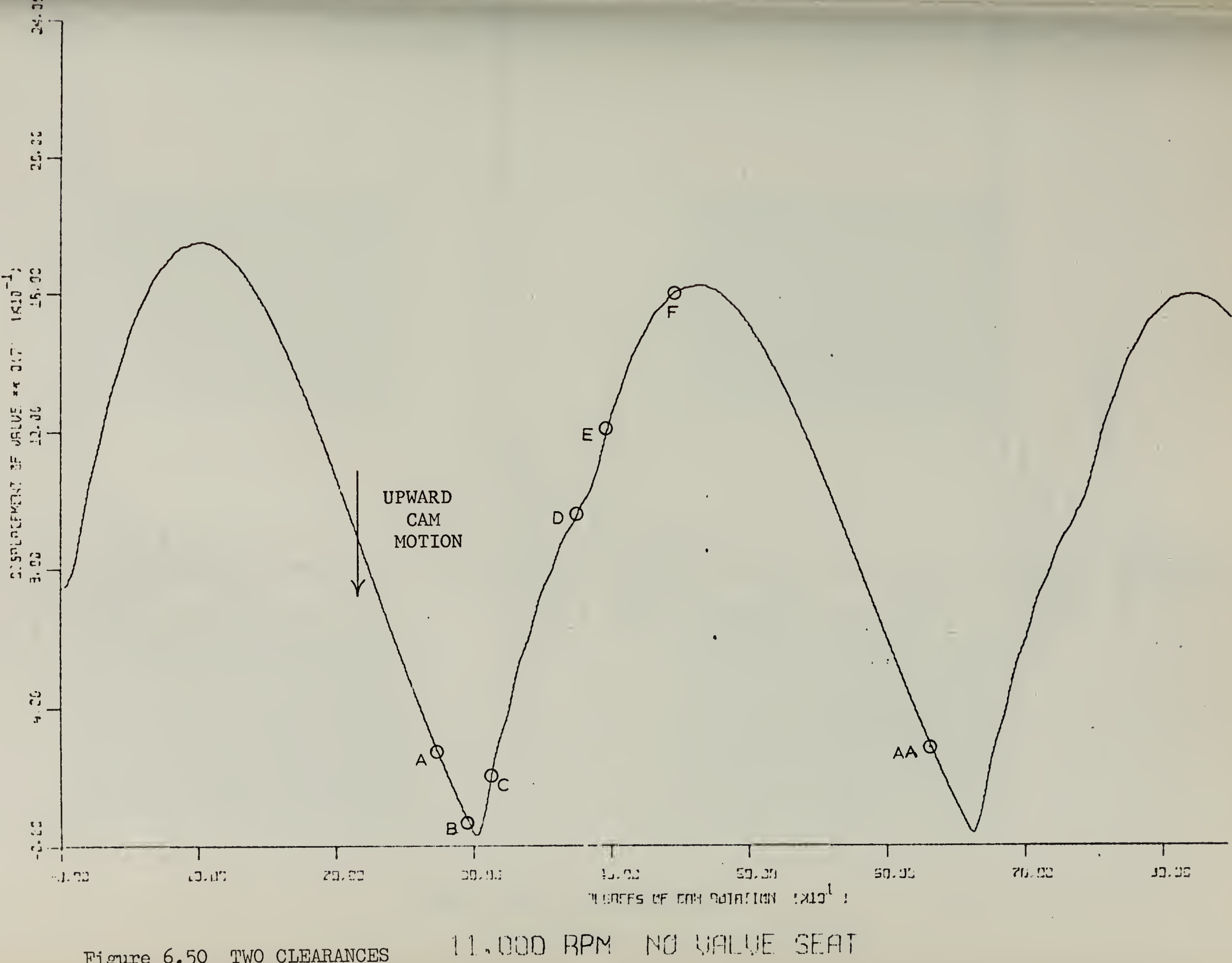


Figure 6.50 TWO CLEARANCES

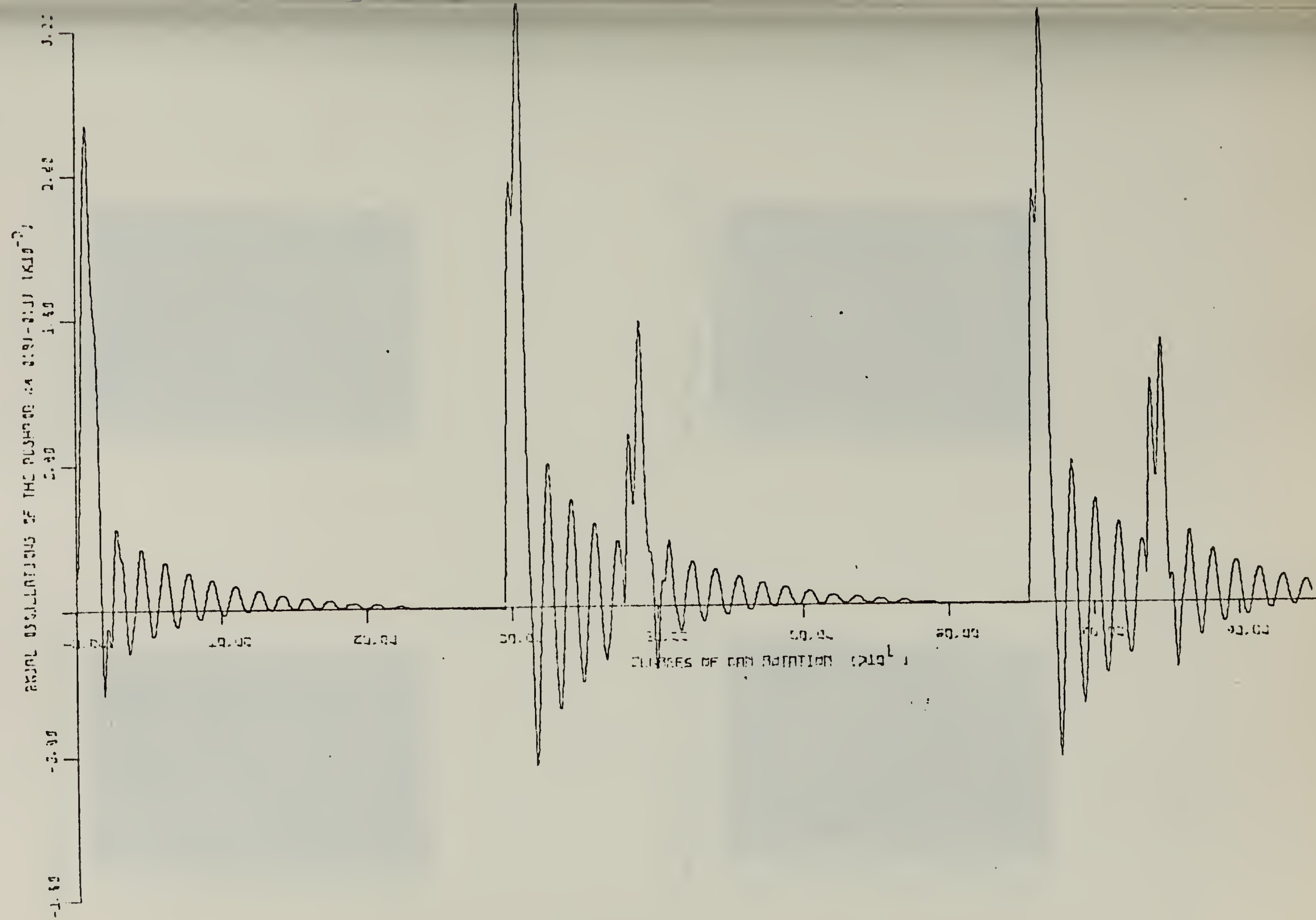


Figure 6.51 TWO CLEARANCES

11,000 RPM NO VALVE SEAT

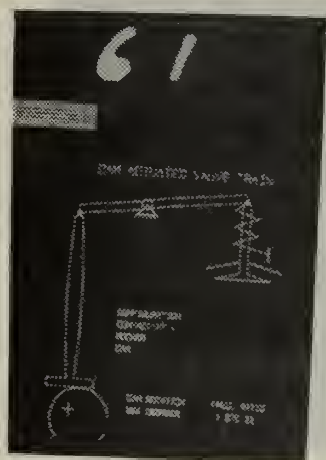


Photo 6.57 - 664 degrees

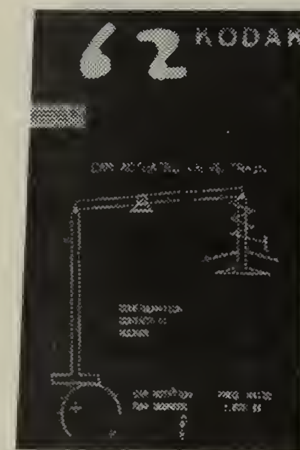


Photo 6.58 - 682 degrees

Photo 6.59 - 694 degrees

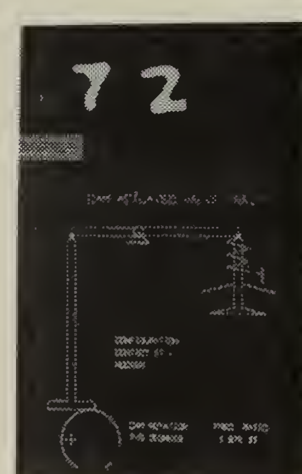


Photo 6.60 - 716 degrees

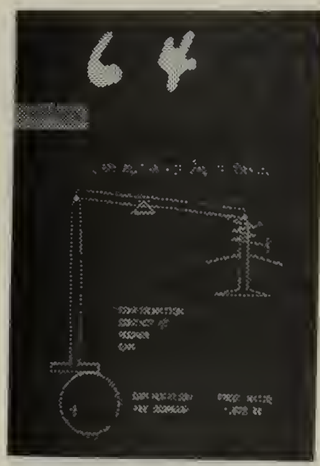


Photo 6.61 - 400 degrees

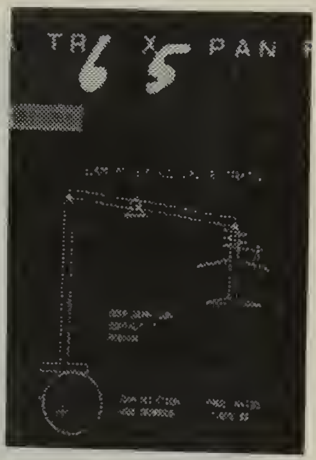


Photo 6.62 - 420 degrees

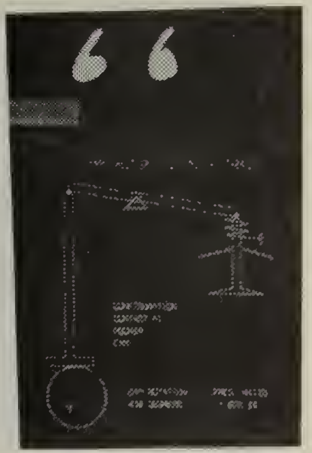


Photo 6.63 - 430 degrees

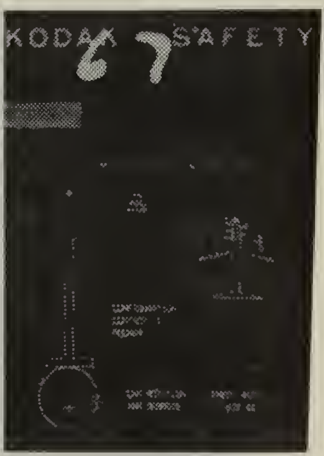


Photo 6.64 - 452 degrees

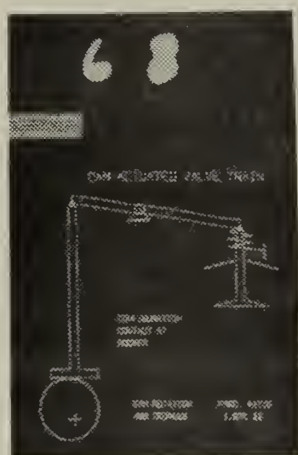


Photo 6.65 - 466 degrees

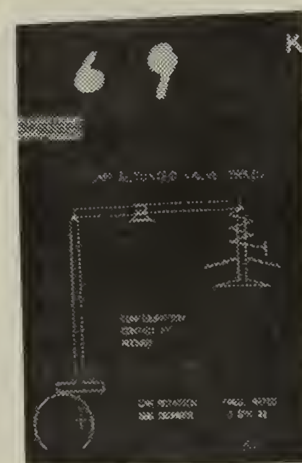


Photo 6.67 - 586 degrees

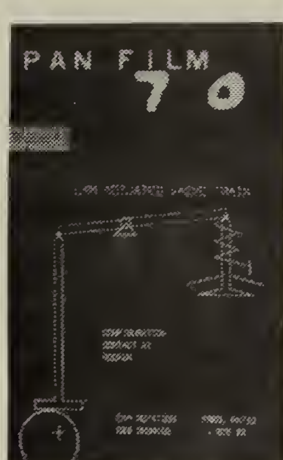


Photo 6.66 - 600 degrees

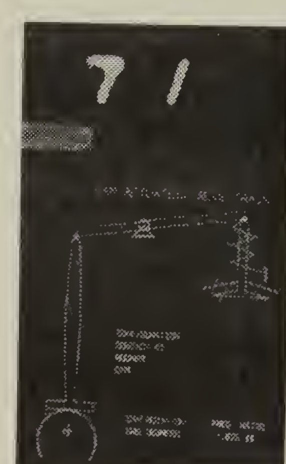


Photo 6.68 - 626 degrees

5. Example two, 9000 RPM and example three, 10,000 RPM were choosen for comparison and completeness of this study. These examples are not annotated and may be found in Appendix F.

VII. CONCLUSION

The simulation for the dynamic response of an elastic link model with three clearances has been described. The algorithm developed is applied to the automobile cam actuated valve train assembly, with external clearances at the valve and cam. The internal clearance at the pushrod-left rocker arm, requires the solution to a second eigenvalue problem in a portion of the calculations. These algorithms are generally coded to allow for application to similar mechanisms. The methods of calculation use address modification arrays to enable application of the program to more complicated models by changing a minimum number of variables and array elements. The numerous subroutines developed have, in general, very specific functions, and are intended to be used as separate calculation packages in a variety of other applications.

While the accuracy of the mathematical model of a mechanism has been greatly increased over the rigid version, the complexity of solution has also increased, even though a number of simplifying assumptions were made. The major assumption made by the theory was that of a linear model so as to permit superposition. Other assumptions, made to simplify the equations but not the theory, involved the choice of coordinates and the use of straight, constant area links.

The purpose of this work was to demonstrate a more accurate model of a mechanism. The extent of this work was to (1) restructure and reprogram an existing simulation to provide for a generalized

method of establishing the physical model configurations; (2) Modify this simulation to allow for more than two clearances; (3) Establish a method for the dynamic graphical display of the model showing the resulting elastic deformations of the model members at high operational speeds. Objectives (1) and (2) are accomplished using the IBM-360 computer for a three clearance simulation. Dynamic dimensioning and program address modifications arrays contribute to the program generalization. Sample problem graphical output is provided. Objective (3) is accomplished with the aid of the ADAGE Graphical Terminal, model 10 interfaced with a Scientific Data System Computer model 9300. Data generated by the IBM computer is tranferred to the SDS computer by punched cards. A program then calculates the model's geometry, position, and deformations. The results are then displayed on the AGT. These results are provided on video tape and may be viewed by requesting NPGS Video Tape 12NPS7200042-VR from the Educational Media Department, Naval Postgraduate School.

It is suggested that additional study be done on the calculation of the pushrod bounce at the rocker. The simulation, as written, has difficulty in obtaining a stable system when contact is regained at this point.

An extension of the AGT system would be to allow for the display of force and displacement curves on the AGT. Methods similar to those developed by this author and Lt. Bowden (Ref. 19) could be conveniently applied.

APPENDIX A

MASS AND STIFFNESS MATRICES

The system mass and stiffness matrices are given by equations (3.6) and (3.7), and are repeated below.

$$[M] = \sum_{i=1}^{5} [\beta]_i^T \, [M_e]_i \, [\beta]_i \tag{3.6}$$

$$[K] = \sum_{i=1}^{5} [\beta]_i^T \, [K_e]_i \, [\beta]_i \tag{3.7}$$

The transformantion matrix $[\beta]$ is given in equation (3.4) and will not be repeated here. The element stiffness and mass matrices are given by reference (1). For the axial members (pushrod, spring, and valve) they are:

$$[K_e]_i = \frac{E_i \, A_i}{L_i} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{A.1}$$

$$[M_e]_i = \frac{M_i}{6} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \qquad (i=1,4,5) \tag{A.2}$$

The element stiffness and mass matrices for the bending members (left and right arms) are:

$$[K_e]_i = \frac{E_i \, I_i}{L_i^3} \begin{bmatrix} 12 & -12 & 6L_i & 6L_i \\ -12 & 12 & -6L_i & -6L_i \\ 6L_i & -6L_i & 4L_i^2 & 2L_i^2 \\ 6L_i & -6L_i & 2L_i^2 & 4L_i^2 \end{bmatrix} \tag{A.3}$$

$$[M_e]_i = \frac{M_i}{420} \begin{bmatrix} 156 & 54 & 22L_i & -13L_i \\ 54 & 156 & 13L_i & -22L_i \\ 22L_i & 13L_i & 4L_i^2 & -3L_i^2 \\ -13L_i & -22L_i & -3L_i^2 & 4L_i^2 \end{bmatrix} \quad (i=2,3) \qquad \text{(A.4)}$$

For equations A.1 through A.4, E_i is the element elastic modulus, A_i is the element cross-sectional area, L_i is the element length, I_i is the element moment of inertia about the bending axis, and M_i is the total element mass of the i^{th} member.

Evaluating equation (3.6) for the five elements of the system yields the full system mass matrix, designated as layer one. The elements of this matrix are given by Figure A.1. Similarly, evaluating equation (3.7) yields the system stiffness matrix. The elements of this matrix are shown in Figure A.2.

$$
\begin{bmatrix}
\frac{M_1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{M_1}{6} \\
0 & \frac{156\ M_2}{420} & \frac{22M_2L_2}{420} & 0 & 0 & \frac{-13L_2M_2}{420} & 0 & 0 \\
0 & \frac{22L_2M_2}{420} & \frac{4L_2^2M_2}{420} & 0 & 0 & \frac{-3L_2^2M_2}{420} & 0 & 0 \\
0 & 0 & 0 & \frac{156M_3}{420}+\frac{M_4}{3}+\frac{M_5}{3} & \frac{-22L_3M_3}{420} & \frac{13L_3M_3}{420} & \frac{-M_5}{6} & 0 \\
0 & 0 & 0 & \frac{-22L_3M_3}{420} & \frac{4L_3^2M_3}{420} & \frac{-3L_3^2M_3}{420} & 0 & 0 \\
0 & \frac{-13L_2M_2}{420} & \frac{-3L_2^2M_2}{420} & \frac{13L_3M_3}{420} & \frac{-3L_3^2M_3}{420} & \frac{4L_2^2M_2}{420} & 0 & 0 \\
0 & 0 & 0 & \frac{-M_5}{6} & 0 & 0 & \frac{M_5}{3} & 0 \\
\frac{M_1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{M_1}{3}
\end{bmatrix}
$$

Figure A.1 Full System Mass Matrix

$$
\begin{bmatrix}
\frac{E_1A_1}{L_1} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-E_1A_1}{L_1} \\
0 & \frac{+12E_2I_2}{L_2^3} & \frac{6E_2I_2}{L_2^2} & 0 & 0 & \frac{6E_2I_2}{L_2^2} & 0 & 0 \\
0 & \frac{6E_2I_2}{L_2^2} & \frac{4E_2I_2}{L^2} & 0 & 0 & \frac{2E_2I_2}{L_2} & 0 & 0 \\
0 & 0 & 0 & \begin{matrix} \frac{12E_3I_3}{L_3^3} \\ \frac{+E_5A_5}{L_5} \\ \frac{+E_4A_4}{L_4} \end{matrix} & \frac{-6E_3I_3}{L_3^2} & \frac{-6E_3I_3}{L_3^2} & \frac{E_5A_5}{L_5} & 0 \\
0 & 0 & 0 & \frac{-6E_3I_3}{L_3^2} & \frac{4E_3I_3}{L_3} & \frac{2E_3I_3}{L_3} & 0 & 0 \\
0 & \frac{6E_2I_2}{L_2^2} & \frac{2E_2I_2}{L_2} & \frac{-6E_3I_3}{L_3^2} & \frac{2E_3I_3}{L_3} & \begin{matrix} \frac{4E_3I_3}{L_3} \\ \frac{+4E_2I_2}{L_2} \end{matrix} & 0 & 0 \\
0 & 0 & 0 & \frac{E_5A_5}{L_5} & 0 & 0 & \frac{E_5A_5}{L_5} & 0 \\
\frac{-E_1A_1}{L_1} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{E_1A_1}{L_1}
\end{bmatrix}
$$

Figure A.2 Full System Stiffness Matrix

The mass and stiffness matrices are valid for the eight coordinate system, i.e. configuration I, II, V, and VI. When the rocker is in contact with the pushrod, the seven coordinate system is used and the transformation, mass and stiffness revert to those developed by Anderson, reference (2). The reduced system mass and stiffness matrices are shown in Figures A.3 and A.4, respectfully. These reduced matrices are stored in layer 3 and are used for configurations III, IV, VII, and VIII.

$$
\begin{bmatrix}
\frac{M_1}{3} & \frac{-M_1}{6} & 0 & 0 & 0 & 0 & 0 \\[2ex]
\frac{-M_1}{6} & \frac{M_1}{3} + \frac{156M_2}{420} & \frac{22M_2L_2}{420} & 0 & 0 & \frac{-13L_2M_2}{420} & 0 \\[2ex]
0 & \frac{22L_2M_2}{420} & \frac{4L_2^2M_2}{420} & 0 & 0 & \frac{-3L_2^2M_2}{420} & 0 \\[2ex]
0 & 0 & 0 & \begin{matrix}\frac{156M_3}{420} + \frac{M_4}{3} \\ +\frac{M_5}{3}\end{matrix} & \frac{-22L_3M_3}{420} & \frac{13L_3M_3}{420} & \frac{-M_5}{6} \\[2ex]
0 & 0 & 0 & \frac{-22L_3M_3}{420} & \frac{4L_3^2M_3}{420} & \frac{-3L_3^2M_3}{420} & 0 \\[2ex]
0 & \frac{-13L_2M_2}{420} & \frac{-3L_2^2M_2}{420} & \frac{13L_3M_3}{420} & \frac{-3L_3^2M_3}{420} & \frac{4L_2^2M_2}{420} & 0 \\[2ex]
0 & 0 & 0 & \frac{-M_5}{6} & 0 & 0 & \frac{M_5}{3}
\end{bmatrix}
$$

Figure A.3 Reduced System Mass Matrix

$$
\begin{bmatrix}
\frac{E_1A_1}{L_1} & \frac{E_1A_1}{L_1} & 0 & 0 & 0 & 0 & 0 \\
\frac{E_1A_1}{L_1} & \frac{E_1A_1}{L_1} + \frac{12E_2I_2}{L_2^{\,3}} & \frac{6E_2I_2}{L_2^{\,2}} & 0 & 0 & \frac{6E_2I_2}{L_2^{\,2}} & 0 \\
0 & \frac{6E_2I_2}{L_2^{\,2}} & \frac{4E_2I_2}{L_2} & 0 & 0 & \frac{2E_2I_2}{L_2} & 0 \\
0 & 0 & 0 & \frac{12E_3I_3}{L_3^{\,3}} + \frac{E_5A_5}{L_5} + \frac{E_4A_4}{L_4} & \frac{-6E_3I_3}{L_3^{\,2}} & \frac{-6E_3I_3}{L_3^{\,2}} & \frac{E_5A_5}{L_5} \\
0 & 0 & 0 & \frac{-6E_3I_3}{L_3^{\,2}} & \frac{4E_3I_3}{L_3} & \frac{2E_3I_3}{L_3} & 0 \\
0 & \frac{6E_2I_2}{L_2^{\,2}} & \frac{2E_2I_2}{L_2} & \frac{-6E_3I_3}{L_3^{\,2}} & \frac{2E_3I_3}{L_3} & \frac{4E_3I_3}{L_3} + \frac{4E_2I_2}{L_2} & 0 \\
0 & 0 & 0 & \frac{E_5A_5}{L_5} & 0 & 0 & \frac{E_5A_5}{L_5}
\end{bmatrix}
$$

Figure A.4 Reduced System Stiffness Matrix

APPENDIX B

NEWMARK'S β PARAMETER METHOD

The one step algorithm developed by Newmark (Ref. 4), is chosen as the numerical integration technique used in the simulation. The β parameter may be changed for an analysis. This time marching technique solves the governing equations of motion by using the following relations:

$$\{\dot{q}\}^{n+1} = \{\dot{q}\}^n + \frac{\Delta T}{2}\left(\{\ddot{q}\}^n + \{\ddot{q}\}^{n+1}\right) \tag{B.1}$$

and

$$\left.\begin{aligned}\{q\}^{n+1} = \{q\}^n + \Delta T\,\{\dot{q}\}^n + \frac{(\Delta T)^2}{2}\,\{\ddot{q}\}^n \\ + \beta\,(\Delta T)^2\left(\{\ddot{q}\}^{n+1} - \{q^n\}\right)\end{aligned}\right| \tag{B.2}$$

The net effect of the parameter β is to change the form of the variation of acceleration during the time interval ΔT. For this study, $\beta = 1/6$, assumes that the acceleration changes linearly from $\{\ddot{q}\}^n$ to $\{\ddot{q}\}^{n+1}$, Figure B.1.

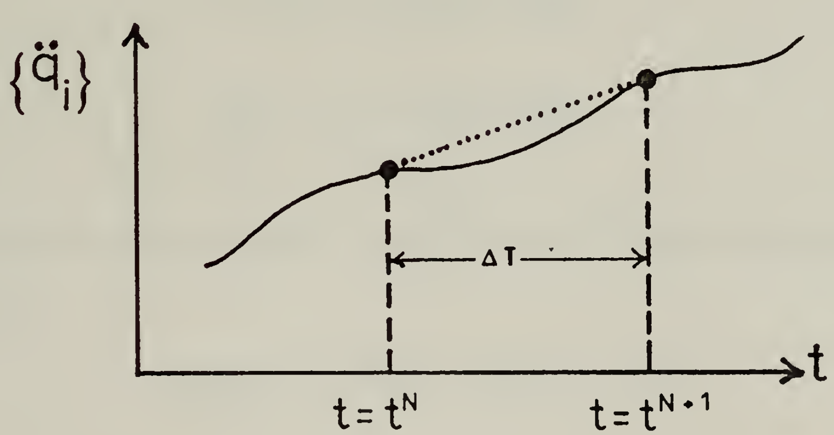


Figure B.1 Time Step Acceleration Curve

[illegible]

[illegible]

In this regard equations B.1 and B.2 are written as:

$$\text{Acceleration} = \ddot{q}_i(t) = \ddot{q}_i^{n} + \frac{(\ddot{q}_i^{n+1} - \ddot{q}_i^{n})}{\Delta T}\, t \tag{B.3}$$

$$\text{Velocity} = \dot{q}_i(t) = \dot{q}_i^{n} + \ddot{q}_i^{n} t + \frac{(\ddot{q}_i^{n+1} - \ddot{q}_i^{n})}{\Delta T}\, \frac{t^2}{2} \tag{B.4}$$

$$\text{Displacement} = q_i(t) = q_i^{n} + \dot{q}_i^{n} t + \ddot{q}_i^{n} \frac{2}{2} + \frac{(\ddot{q}_i^{n+1} - \ddot{q}_i^{n})}{\Delta T}\, \frac{t^3}{6} \tag{B.5}$$

Evaluate $\dot{q}_i(t)$ and $q_i(t)$ at time $t = \Delta T$,

$$\dot{q}_i^{n+1} = \dot{q}_i^{n} + \ddot{q}_i^{n} \frac{\Delta T}{2} + \ddot{q}_i^{n+1} \frac{\Delta T}{2} \tag{B.6}$$

$$q_i^{n+1} = q_i^{n} + \dot{q}_i^{n} \Delta T + \ddot{q}_i^{n} \frac{(\Delta T)^2}{3} + \ddot{q}_i^{n+1} \frac{(\Delta T)^2}{6} \tag{B.7}$$

Let $$\alpha_i^{n} = \dot{q}_i^{n} + \ddot{q}_i^{n} \frac{\Delta T}{2} \tag{B.8}$$

and $$\Psi_i^{n} = q_i^{n} + \dot{q}_i^{n} \Delta T + \ddot{q}_i^{n} \frac{(\Delta T)^2}{3} \tag{B.9}$$

Now $\dot{q}_i^{n+1}$ and q_i^{n} can be rewritten:

$$\dot{q}_i^{n+1} = \alpha_i^{n} + \ddot{q}_i^{n+1} \frac{\Delta T}{2} \tag{B.10}$$

$$q_i^{n+1} = \Psi_i^{n} + \ddot{q}_i^{n+1} \frac{(\Delta T)^2}{6} \tag{B.11}$$

Substitute the above results for $\dot{q}_i^{n+1}$ and q_i^{n+1} into the equations of motion

$$[M]\,\{\ddot{q}\} + C\,\{\dot{q}\} + [K]\,\{q\} = \{F\} \tag{B.12}$$

$$[M]\{\ddot{q}\}^{n+1} + [C]\{\alpha\}^n + \frac{\Delta T}{2}[C]\{\ddot{q}\}^{n+1} + [K]\{\Psi\}^n + \frac{(\Delta T)^2}{6}[K]\{\ddot{q}\}^{n+1}$$

$$= \{P\}^{n+1} \tag{B.13}$$

Rearranged, $$\left[[M] + \frac{\Delta T}{2}[C] + \frac{\Delta T^2}{6}[K]\right]\{\ddot{q}\}^{n+1} = \{P\}^{n+1} - [C]\{\alpha\}^n - [K]\{\Psi\}^n \tag{B.14}$$

Rewritten, $$[\overline{M}]^{n+1}\{\ddot{q}\}^{n+1} = \{\overline{P}\}^{n+1} \tag{B.15}$$

where $\{P\}^{n+1} = \{F\}$ evaluated at time n+1

$$[\overline{M}]^{n+1} = [M] + \frac{\Delta T}{2}[C] + \frac{\Delta T^2}{6}[K] \tag{B.16}$$

$$\{\overline{P}\}^{n+1} = \{P\}^{n+1} - [C]\{\alpha\}^n - [K]\{\Psi\}^n \tag{B.17}$$

This results in the accelerations $\{\ddot{q}\}^{n+1}$ expressed in terms of the known system values at the prior time step, $t = t^n$.

$$\{\ddot{q}\}^{n+1} = \left[[\overline{M}]^{n+1}\right]^{-1}\{\overline{P}\}^{n+1}$$

APPENDIX C

FOUR CLEARANCE PROBLEM

During the early stages of this study, limited research was conducted on the four clearance problem to better understand the necessary requirements of program generalization procedures. This appendix will present the unique four clearance number scheme that must be used in conjunction with the shifting algorithm presented in section 3.D. Figure C.1 indicates the configuration number associated with the various combinations of clearances, for a given time, that the model is physically permitted to take. Figure C.2 shows the resulting flow diagram. The shift lines relate the flow for a nonpositive force at the clearance indicated. Non-positive displacement differences are along the same shift lines, but in the direction opposite the force arrow.

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

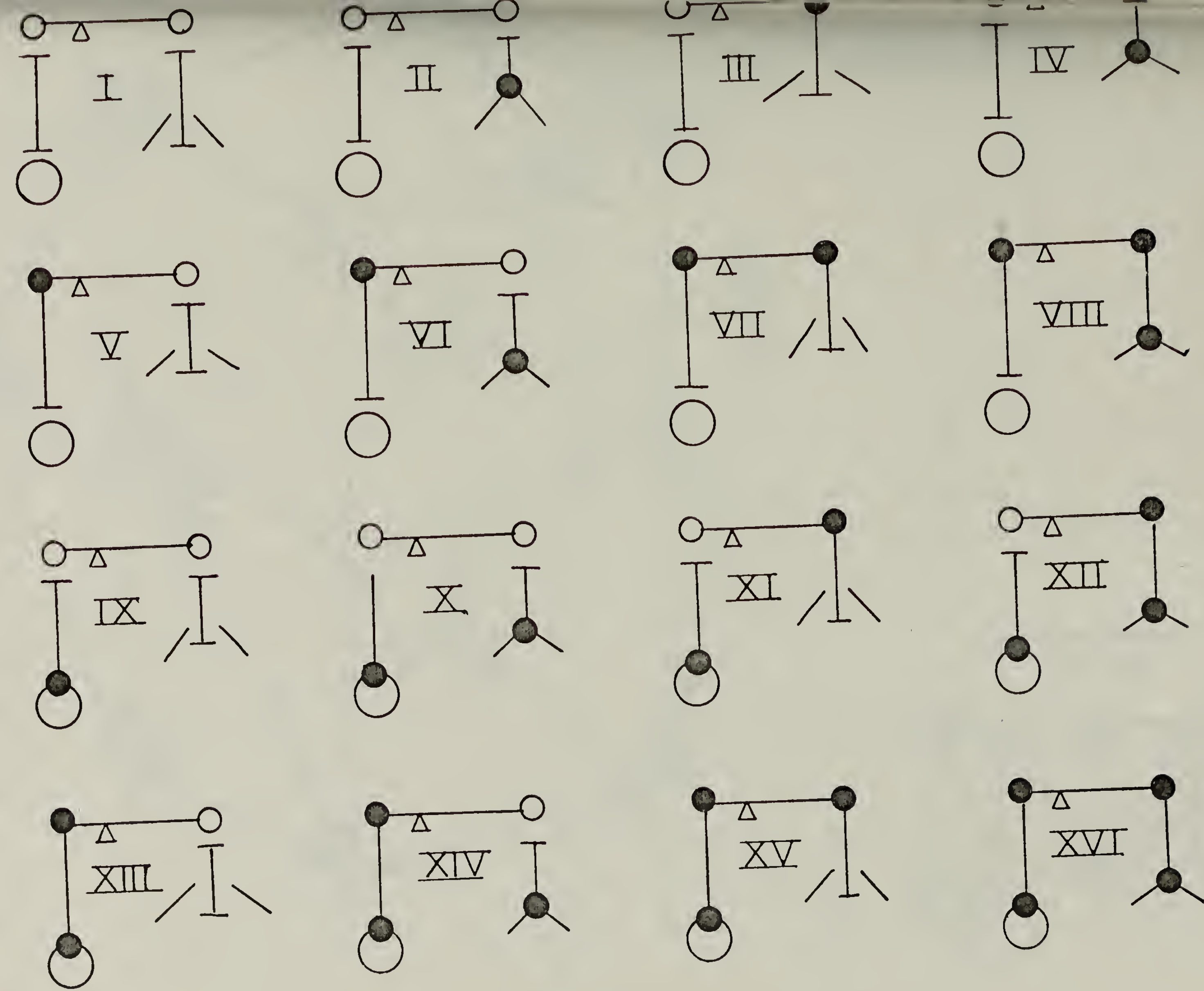
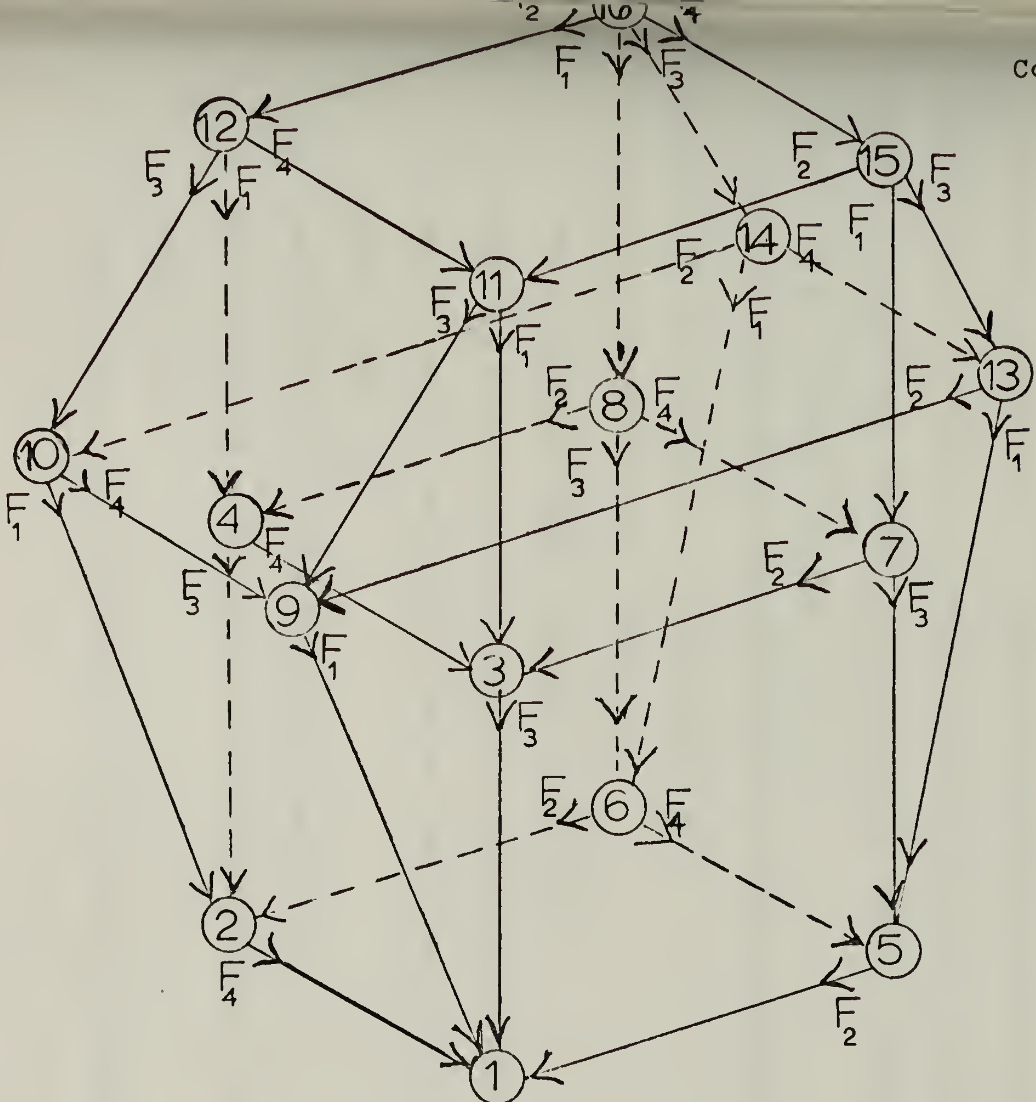


Figure C .1 FOUR CLEARANCE SYSTEM CONFIGURATIONS

Configuration Numbers
Indicated



Clearance Numbers
Indicated

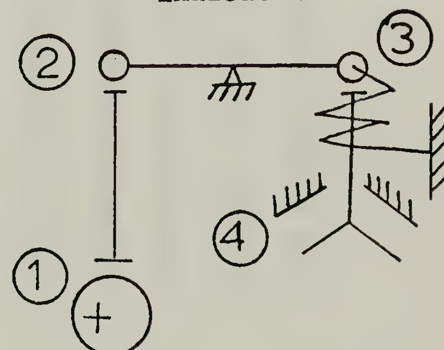


Figure C.2 4 Clearance Configuration Flow Diagram

APPENDIX D

COMPUTER CONTROL CARDS

A. IBM-360 COMPUTER CONTROL CARDS

As previously noted subroutines were complied and placed on a 2314 Disk Storage Device. This accomplished two goals. First, it eliminated the requirement to physically read into the computer a large number of punched cards. Second, it significantly reduced program execution time by greatly reducing the FORT step. This allowed the program to execute in a higher job class.

The following figures list the control cards required to accomplish various disk manipulations. A complete explaination will not be attempted here, but merely a listing of procedures that have been found to be effective. Section 3.4 of the Users Manual, reference (13), should be reviewed for more information.

In the following example control cards, the data set name is S0274. NILKA and is placed on public disk Mary.

1. Initialize disk space, Figure D.1.
2. Load subroutine NEWMRK onto disk, Figure D.2.
3. List disk data set members and space remaining, Figure D.3.
4. Execute program reading only control cards, main, and data cards, Figure D.4.
5. Load main program, CWG, onto disk, Figure D.5.
6. Execute program reading only control cards and data cards, Figure D.6.
7. Compress disk area when available space has been exceeded.

a. Read disk members onto scratch disk without intermediate spacing, Figure D.7.

b. Copy all members to users disk, Figure D.8.

c. Copy all members except NEWMRK to users disk, Figure D.9.

```
//    JOB CARD  (GREEN CARD)
// EXEC   FORTHCL,PARM.LINK='NCAL,LET'
//FORT.SYSIN DD *
--- SUBROUTINE NEWMRK CARDS ---
       STEP DELIMINATOR /* CARD
//LINK.SYSLMOD DD UNIT=2314,VOL=SER=MARY,DSN=S0274.NILKA,
//      DISP=(NEW,KEEP),SPACE=(CYL,(6,1,6)),LABEL=EXPDT=71365
//LINK.SYSIN DD *
   NAME     NEWMRK(R)
  END OF JOB /*            ( ORANGE CARD )
```

Figure D.1 Initalize Disk Space

```
//    JOB CARD  (GREEN CARD)
// EXEC   FORTHCL,PARM.LINK='NCAL,LET'
//FORT.SYSIN DD *
--- SUBROUTINE NEWMRK CARDS ---
       STEP DELIMINATOR /* CARD
//LINK.SYSLMOD DD UNIT=2314,VOL=SER=MARY,DSN=S0274.NILKA,
//      DISP=OLD,SPACE=(CYL,(6,1,6))
//LINK.SYSIN DD *
   NAME     NEWMRK(R)
  END OF JOB /*            ( ORANGE CARD )
```

Figure D.2 Load Subroutine onto Disk

```
//    JOB CARD  (GREEN CARD)
//LIST EXEC PGM=IEHLIST,REGION=58K
//SYSPRINT DD SYSOUT=A,SPACE=(TRK,(1,1))
//DD1 DD UNIT=2314,VOL=SER=MARY,DISP=OLD
//DD2 DD UNIT=2314,VOL=SER=MARY,DISP=OLD
//SYSIN DD *,DCB=BLKSIZE=80
       LISTVTOC FORMAT,VOL=2314=MARY,DSNAME=(S0274.NILKA)
       LISTPDS DSNAME=S0274.NILKA,VOL=2314=MARY
  END OF JOB /*            ( ORANGE CARD )
```

Figure D.3 List Data Set VTOC and PDS

```
//    JOB CARD  (GREEN CARD)
// EXEC   FORTHCLG,REGION.GO=250K
//FORT.SYSIN DD *
----   MAIN PROGRAM CARDS ---
        STEP DELIMINATOR /* CARD
//LINK.PLIB   DD   DSN=S0274.NILKA,VOL=SER=MARY,DISP=SHR,UNIT=2314
//LINK.SYSIN   DD   *
    INCLUDE PLIB(BRANCH,CAM,DEBUG,DEL,DRUP,DVVS,MCK)
    INCLUDE PLIB(INITAL,MATRIX,NEWMRK,OLPLOT,SHIFT)
    INCLUDE PLIB(BKSTEP)
    INCLUDE PLIB(INPUT)
    INCLUDE PLIB(GRAPH)
    INCLUDE PLIB(AGT)
    INCLUDE PLIB(THESIS)
    INCLUDE PLIB(PRINT)
    INCLUDE PLIB(ARRAY)
        STEP DELIMINATOR /* CARD
//GO.FT06F001 DD SYSOUT=A,SPACE=(CYL,(4,2))
//GO.SYSIN DD *
---  DATA CARDS ---
  END OF JOB /*             ( ORANGE CARD )
```

Figure D.4 Execute Program with Subroutines on Disk

```
//    JOB CARD  (GREEN CARD)
//   EXEC   FORTHCL,PARM.FORT='NAME=CWG,LOAD,MAP',
//        PARM.LINK='LET,NCAL,MAP,XREF'
//FORT.SYSIN DD *
---  MAIN PROGRAM CARDS ---
        STEP DELIMINATOR /* CARD
//LINK.SYSLMOD DD UNIT=2314,VOL=SER=MARY,DSN=S0274.NILKA,
//      DISP=OLD,SPACE=(CYL,(6,1,6))
//LINK.SYSIN DD *
  NAME    CWG(R)
  END OF JOB /*             ( ORANGE CARD )
```

Figure D.5 Load Main Program onto Disk

```
//    JOB CARD  (GREEN CARD)
//  EXEC LGOP,REGION.GO=250K
//LINK.PLIB DD UNIT=2314,VOL=SER=MARY,DSN=S0274,NILKA,DISP=SHR
//LINK.SYSIN   DD   *
  INCLUDE PLIB(ARRAY,BKSTEP,BRANCH,CAM,DEBUG,DEL,DRUP,DVVS,GRAPH,MCK)
  INCLUDE PLIB(INITAL,INPUT,MATRIX,NEWMRK,OLPLOT,PRINT,SHIFT,THESIS)
  INCLUDE PLIB(CWG)
   ENTRY CWG
       STEP DELIMINATOR /* CARD
//GO.SYSIN   DD   *
---  DATA CARDS ---
 END OF JOB /*             ( ORANGE CARD )
```

Figure D.6 Execute Entire Program from Disk

```
//   JOB CARD  (GREEN CARD)
//STEP1  EXEC  PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//D5 DD UNIT=2314,VOL=(PRIVATE,,SER=MARY),DISP=SHR
//SYSIN     DD  *
      LISTVTOC   VOL=2314=MARY
//STEP2  EXEC  PGM=IEHPROGM
//SYSPRINT DD  SYSOUT=A
//SPOOL3   DD  DISP=OLD,UNIT=2314,VOLUME=SER=SPOOL3
//TMPLIB   DD  DISP=(NEW,KEEP),UNIT=2314,DSNAME=NILKA.TMPLIB,
//             SPACE=(CYL,(6,0,17)),DCB=(RECFM=U,BLKSIZE=7294),
//             VOLUME=SER=SPOOL3
//SYSIN    DD  *
  SCRATCH  DSNAME=S0274.NILKA,VOL=2314=SPOOL3,PURGE
  RENAME   DSNAME=NILKA.TMPLIB,VOL=2314=SPOOL3,NEWNAME=S0274.NILKA
     STEP DELIMINATOR /* CARD
//STEP3  EXEC  PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD  DISP=OLD,UNIT=2314,DSNAME=S0274.NILKA,VOLUME=SER=MARY
//SYSUT2   DD  DISP=OLD,UNIT=2314,DSNAME=S0274.NILKA,VOLUME=SER=SPOOL3
//SYSIN    DD  *
  COPY
  END OF JOB /*          ( ORANGE CARD )
```

Figure D.7 Read Data Set Members onto Scratch Disk

```
//   JOB CARD  (GREEN CARD)
//STEP3  EXEC  PGM=IEHPROGM
//SYSPRINT DD  SYSOUT=A
//MARY     DD  DISP=OLD,UNIT=2314,VOLUME=SER=MARY
//SYSIN    DD  *
  SCRATCH  DSNAME=S0274.NILKA,VOL=2314=MARY,PURGE
     STEP DELIMINATOR /* CARD
//STEP4  EXEC  PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD  DISP=OLD,UNIT=2314,DSNAME=S0274.NILKA,VOLUME=SER=SPOOL3
//SYSUT2   DD  DISP=(NEW,KEEP),UNIT=2314,DSNAME=S0274.NILKA,
//             SPACE=(CYL,(6,1,6)),DCB=(RECFM=U,BLKSIZE=7294),
//             VOLUME=SER=MARY,LABEL=EXPDT=71365
//SYSIN    DD  *
  COPY
  END OF JOB /*          ( ORANGE CARD )
```

Figure D.8 Copy all Data Set Members

```
//   JOB CARD  (GREEN CARD)
//STEP3  EXEC  PGM=IEHPROGM
//SYSPRINT DD  SYSOUT=A
//MARY     DD  DISP=OLD,UNIT=2314,VOLUME=SER=MARY
//SYSIN    DD  *
  SCRATCH  DSNAME=S0274.NILKA,VOL=2314=MARY,PURGE
     STEP DELIMINATOR /* CARD
//STEP4  EXEC  PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD  DISP=OLD,UNIT=2314,DSNAME=S0274.NILKA,VOLUME=SER=SPOOL3
//SYSUT2   DD  DISP=(NEW,KEEP),UNIT=2314,DSNAME=S0274.NILKA,
//             SPACE=(CYL,(6,1,6)),DCB=(RECFM=U,BLKSIZE=7294),
//             VOLUME=SER=MARY,LABEL=EXPDT=71365
//SYSIN    DD  *
  COPY   TYPCOPY=E,MAXNAME=1
  MEMBER NAME=(NEWMRK)
  END OF JOB /*          ( ORANGE CARD )
```

Figure D.9 Copy Part of Data Set Members

B. SDS-9300 COMPUTER CONTROL CARDS

Transferring the SDS-9300 program, in Fortran source code, from punched cards to magnetic tape is a relatively easy task. Once accomplished, repeated compiling and execution of the program is made much easier. Figure D.10 indicates the control cards required to read the punched cards onto magnetic tape.

```
 SDS-9300 COMPUTER BINARY BOOT CARD*
¬JOB
¬LOAD X, TAPEDT
$
+
--- MAIN PROGRAM CARDS ---
=EOF
--- SUBROUTINE BAR1 CARDS ---
--- SUBROUTINE BAR2 CARDS ---
--- SUBROUTINE BAR3 CARDS ---
--- SUBROUTINE BAR4 CARDS ---
=EOF
--- SUBROUTINE CAM CARDS ---
=EOF
 ETC.  UNTIL ALL SUBROUTINES ARE READ.
=EOF
```

*Indicates card supplied by Computer Laboratory

Figure D.10 Transferring FORTRAN Source Code From Cards to Magnetic Tape

The computer laboratory provided routine TAPEDT enables the user to change the program coding with a minimum of cards, while the program remains on the tape.

Figure D.11 shows the required control cards to compile, load, and execute the magnetic tape Fortran source program.

```
SDS-9300 COMPUTER BINARY BOOT CARD*
¬JOB
$
¬ASSIGN SI=MT2A
¬REWIND SI
¬DATE 17,DEC,1971
¬TITLE .              CHARLES  W.  GNILKA - -  THESIS - -
¬AGT
¬ASSIGN 4=MT4A
¬LABEL MAIN
¬FORTRAN GO
¬LABEL CAM
¬FORTRAN GO
¬LABEL MIDBAR
¬FORTRAN GO
¬LABEL SPRCTR
¬FORTRAN GO
¬LABEL BAR
¬FORTRAN GO
¬LABEL DISPLY
¬FORTRAN GO
¬LABEL INITAL
¬FORTRAN GO
¬LABEL READ
¬FORTRAN GO
¬LOAD XR,MAP
¬SEG MAIN-BAR-CAM-(MIDBAR,SPRCTR,DISPLY,INITAL,READ)
¬DATA
```

*Indicates Card Supplied by Computer Laboratory

Figure D.11 Program Execution From FORTRAN Object on Magnetic Tape

To place the program on magnetic tape as binary executable code, the control cards are as in Figure D.11 with an additional control card immediately after the Δ JOB card. Figure D.12 indicates the required control cards.

```
 SDS-9300 COMPUTER BINARY BOOT CARD*
¬JOB
$
¬ASSIGN GO=MT3A
  ( THE REMAINING CONTROL CARDS ARE EXACTLY AS GIVEN ABOVE )
```

Figure D.12 Placing Program as Binary Code on Magnetic Tape

Once the binary code is on magnetic it may be executed approximately 75% faster than in uncompiled FORTRAN source form. The control cards to accomplish this are shown in Figure D.13

```
 SDS-9300 COMPUTER BINARY BOOT CARD*
¬JOB
$
¬AGT
¬ASSIGN BI=MT3A
¬ASSIGN GO=BI
¬ASSIGN SI=MT3A
¬LOAD XR,MAP
¬SEG MAIN-BAR-CAM-(MIDBAR,SPRCTR,DISPLY,INITAL,READ)
¬DATA
```

Figure D.13 Program Execution From Binary Code on Magnetic Tape

APPENDIX E

IBM-360 COMPUTER PROGRAM DATA CARDS

```
      GNILKA, C. W. 17 DECEMBER   THREE CLEARANCES  NO. 1
1,000 RPM      VALVE SEAT
1000.              RPM
.7          NO     CLEARANCE AT VALVE SEAT AND PUSHROD/ROCKER
ES               PRINT OUT MATRICS    MPRINT
0              CALCOMP GRAPHS DESIRED
0              AGT OUTPUT
0  NO          PUNCHED CARDS;  MAGNETIC TAPE
.052                   DEGREES TO PRINT DEBUG OUTPUT OR STOP
10    10        PRINTED OUTPUT DESIRED;    NCOUNT
10          PRINTED OUTPUT DESIRED FOR DEBUG PURPOSES - IPRINT
10                OFFLINE PRINTER GRAPH
  1           IDEG - - FORCED TO STAY IN CONFIG 7
).000001               DELT- - TIME INCREMENT
     DISPL. DIFF. BETWEEN ROD & ROCKER ** Q(2)-Q(8)
     DISPL. DIFF. BETWEEN CAM & ROD ** Q(1)-Q(9)
     CAM PROFILE ** Q(9)
     DISPL. OF LOWER END OF PUSHROD ** Q(1)
     FORCE ON PUSHROD ** F(1)
     AXIAL OSCILLATIONS OF THE PUSHROD ** Q(8)-Q(1)
     DISPLACEMENT OF VALVE ** Q(7)
     POSITION OF VALVE SEAT ** Q(10)
     FORCE ON ROCKER ** F(2)
     AXIAL OSCILLATIONS OF THE VALVE STEM
     FORCE ON VALVE ** F(7)
     CAM PROFILE AND DISPL. OF PUSHROD
     DISPL. OF VALVE AND VALVE SEAT
DEGREES OF CAM ROTATION
```

APPENDIX F

This appendix presents two additional examples for comparison with the 11,000 RPM examples presented in chapter 6.C.

A. EXAMPLE 3, THREE CLEARANCES, 9,000 RPM, VALVE SEAT

Approximately 800 degrees of cam rotation is indicated on the associated graphs. These graphs only indicate the indicate the major system points of interest and are not otherwise annotated. The graphs, in order of presentation are:

1. Figure F.1
   a. Cam Profile, $q_9 \times 10^{-1}$
   b. Displacement of lower end of pushrod
      $q_1 \times 10^{-1}$
   c. Displacement difference between cam and pushrod
      $q_1 - q_9 \times 10^{-3}$
2. Figure F.2, Force on Pushrod
   $F_1 \times 10^2$
3. Figure F.3, Force on Rocker
   $F_2 \times 10^2$
4. Figure F.4, Force on Valve
   $F_7 \times 10^2$
5. Figure F.5, Displacement of Valve
   $q_7 \times 10^{-1}$
6. Figure F.6, Axial Oscillations of the Pushrod
   $(q_8 - q_1) \times 10^{-3}$
7. Figure F.7, Axial Oscillation of the Valve Stem
   $(q_7 - q_4) \times 10^{-3}$

8. Figure F.8

   a. Displacement Difference between Cam and Pushrod

      $(q_1 - q_9) \times 10^{-3}$

   b. Displacement Difference between Pushrod and Rocker

      $(q_2 - q_8) \times 10^{-2}$

B. EXAMPLE 3, THREE CLEARANCES 10,000 RPM, VALVE SEAT

This last example referred to in chapter 6.C. also presents approximately 800 degrees of cam rotation. These graphs only indicate the major system points of interest and are not otherwise annotated. The graphs, in order of presentation are:

1. Figure F.9

   a. Cam Profile, $q_9 \times 10^{-1}$

   b. Displacement of lower end of pushrod

      $q_1 \times 10^{-1}$

   c. Displacement difference between cam and pushrod

      $q_1 - q_9 \times 10^{-3}$

2. Figure F.10, Force on Pushrod

   $F_1 \times 10^2$

3. Figure F.11, Force on Rocker

   $F_2 \times 10^2$

4. Figure F.12, Force on Valve

   $F_7 \times 10^2$

5. Figure F.13, Displacement of Valve

   $q_7 \times 10^{-1}$

6. Figure F.14, Axial Oscillations of the Pushrod

   $(q_8 - q_1) \times 10^{-3}$

7. Figure F.15, Axial Oscillation of the Valve Stem

   $(q_7 - q_4) \times 10^{-3}$

8. Figure F.16

a. Displacement Difference between cam and pushrod

$(q_1 - q_9) \times 10^{-3}$

b. Displacement Difference between Pushrod and Rocker

$(q_2 - q_8) \times 10^{-2}$

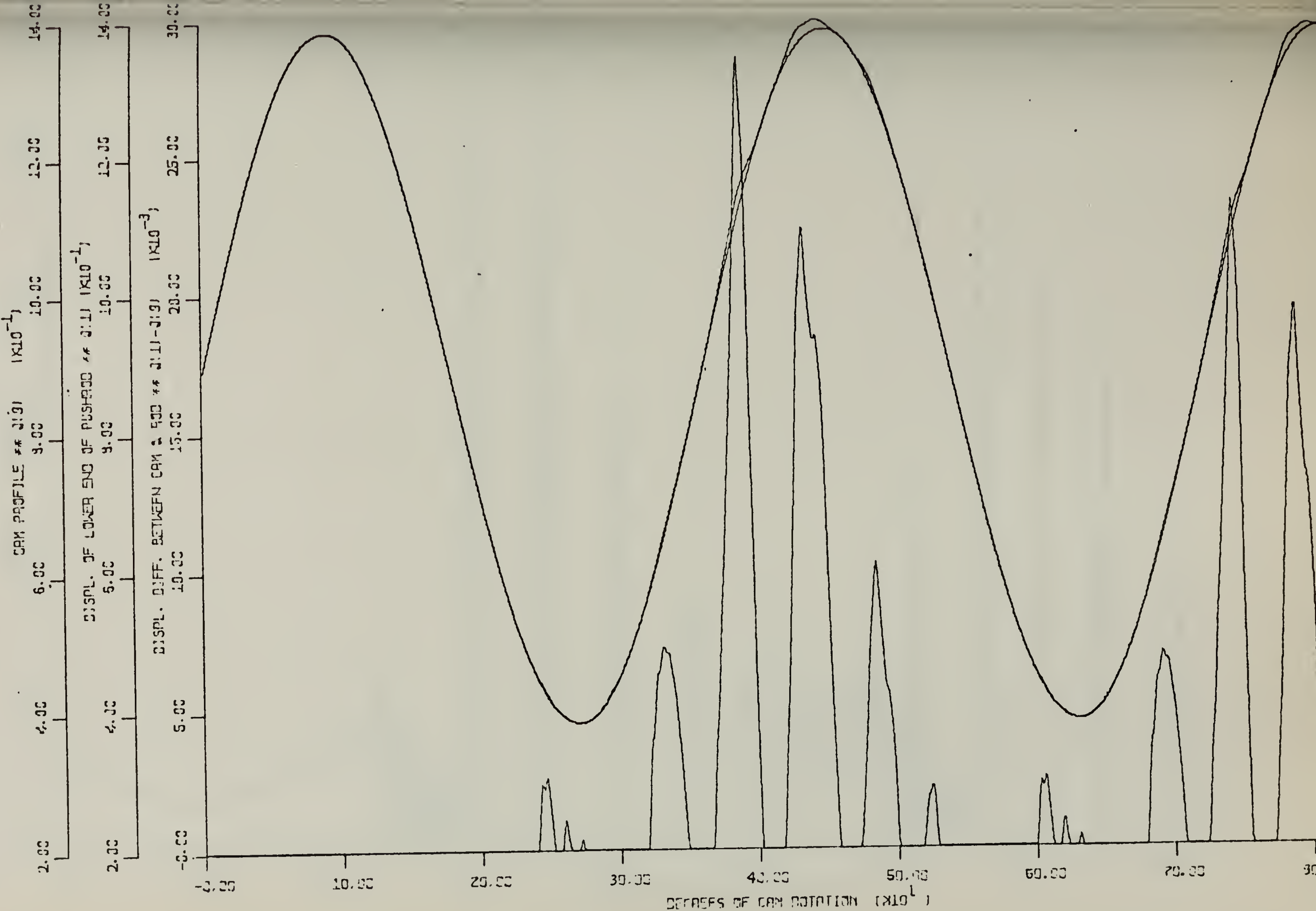


Figure F.1 THREE CLEARANCES 9,000 RPM VALVE SEAT

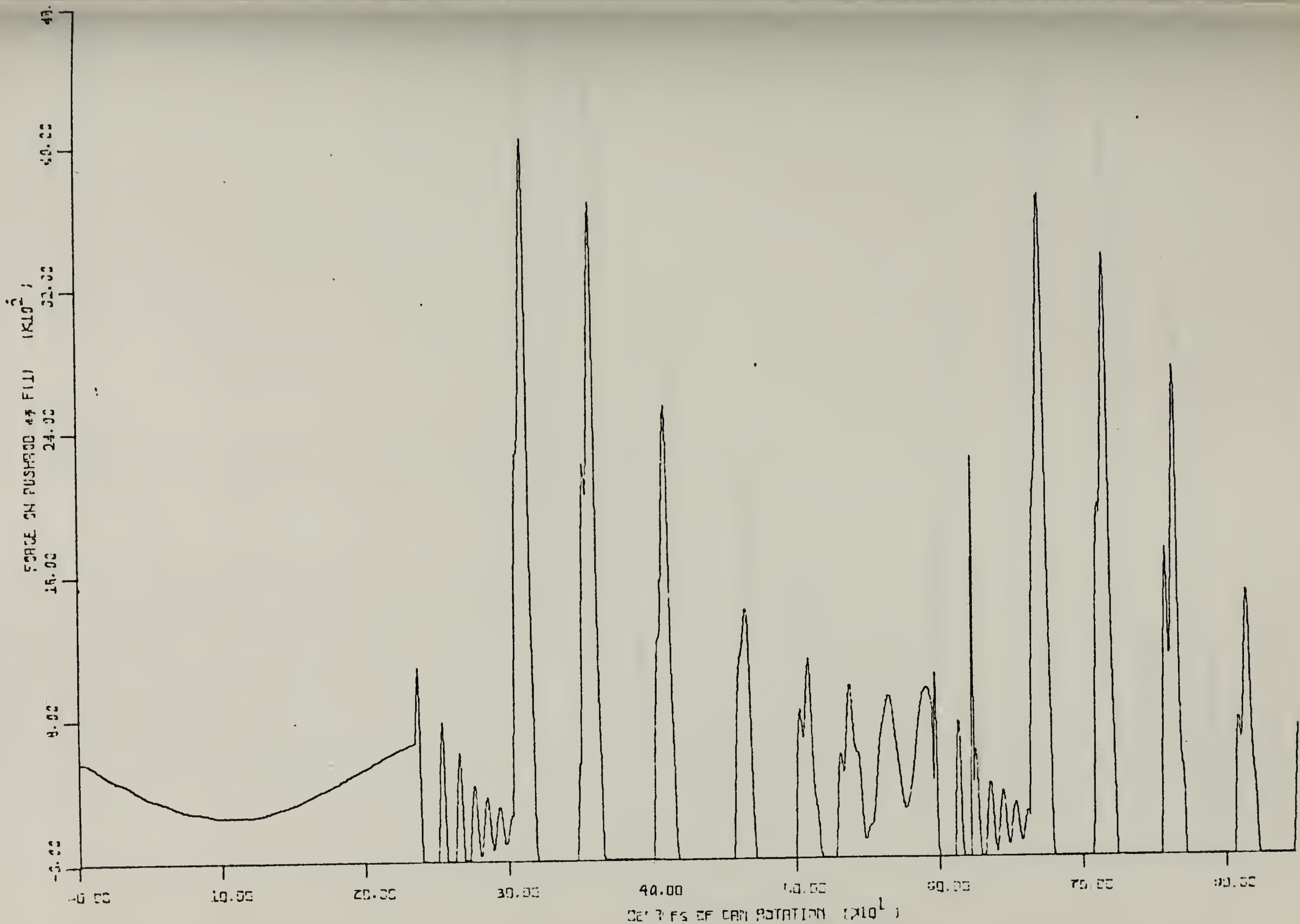


Figure F.2 THREE CLEARANCES 9,000 RPM VALVE SEAT

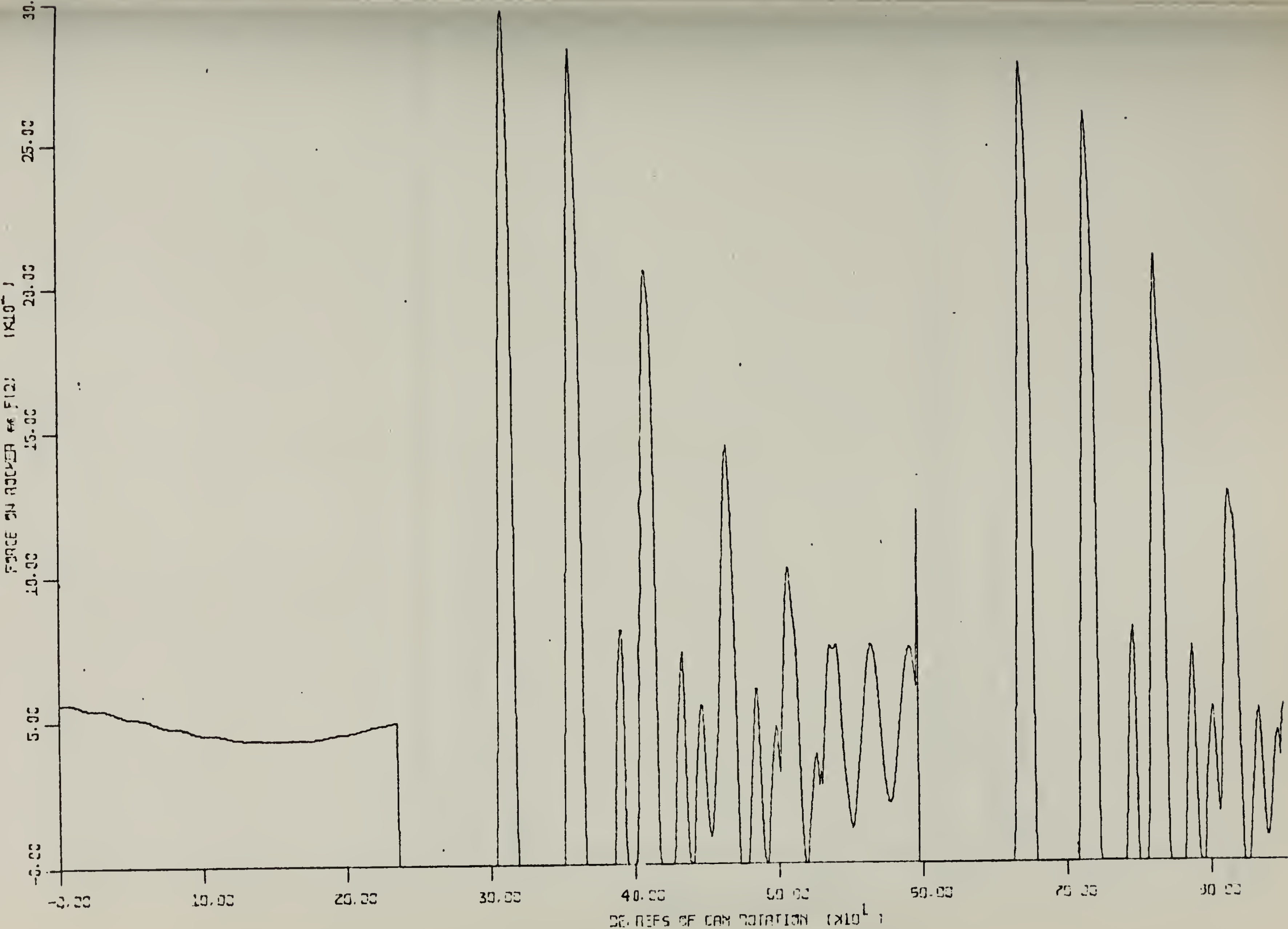
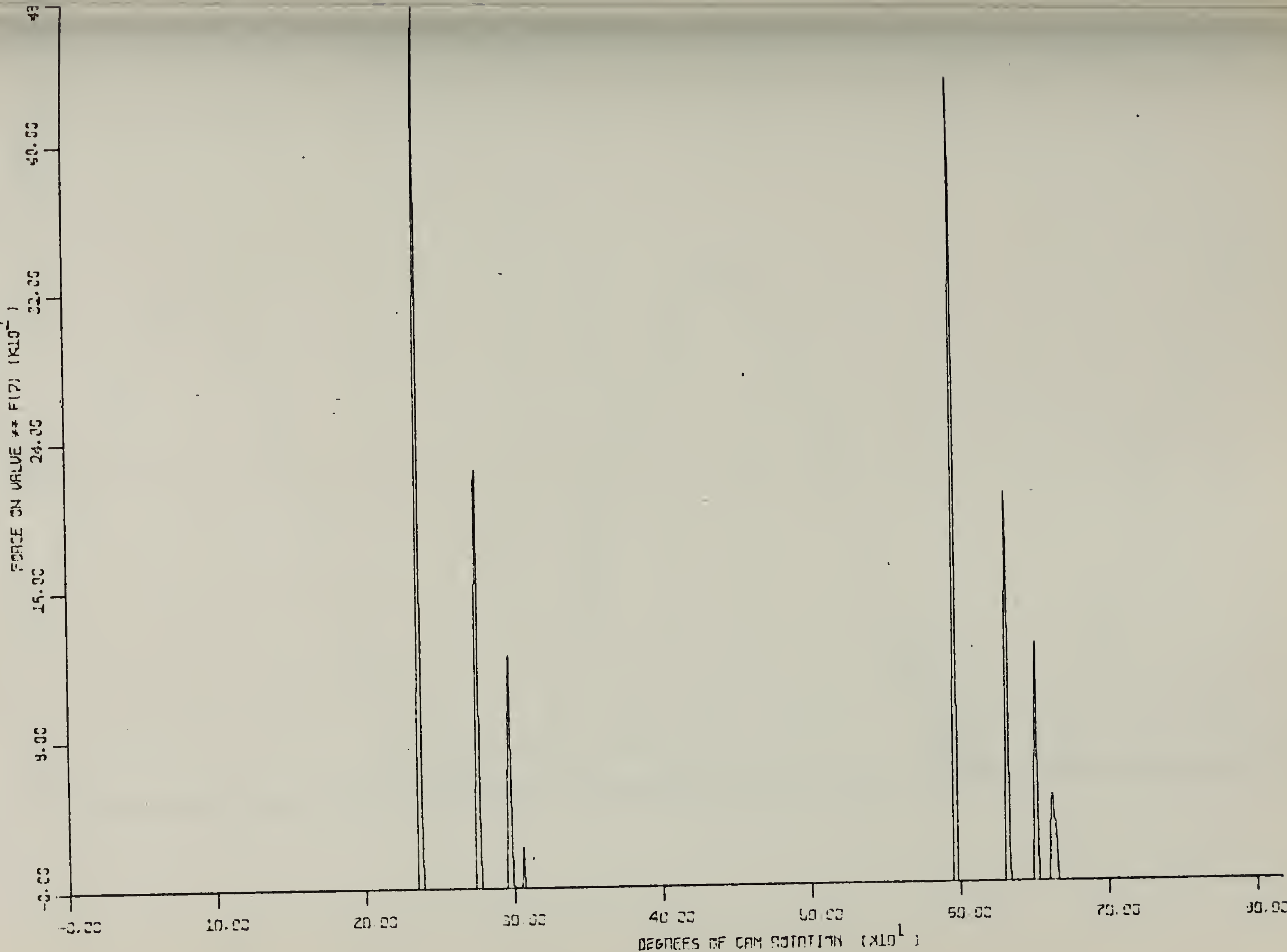


Figure F.3 THREE CLEARANCES 9,000 RPM VALVE SEAT



9,000 RPM VALVE SEAT

Figure F.4 THREE CLEARANCES

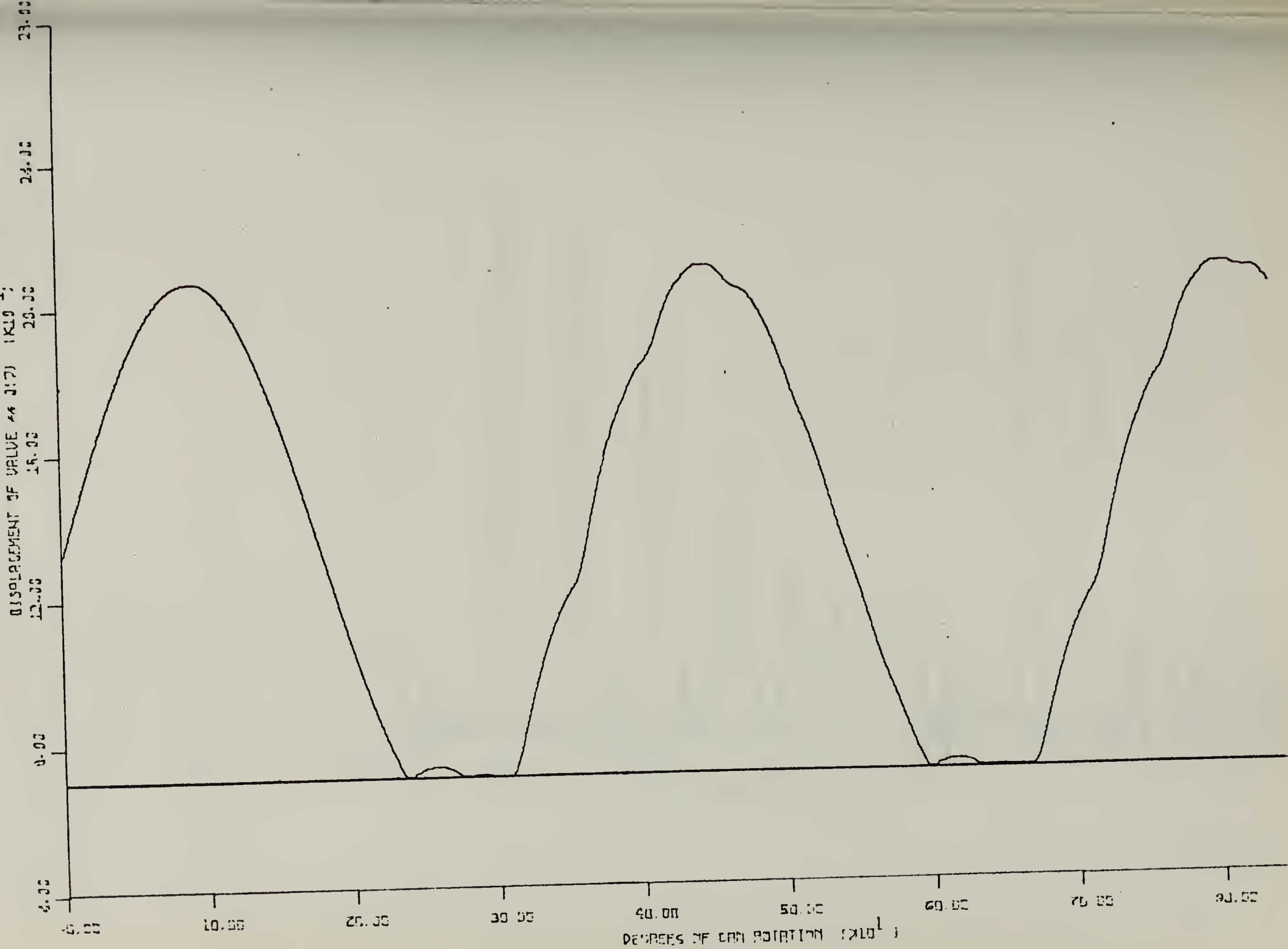


Figure F.5 THREE CLEARANCES 9,000 RPM VALVE SEAT

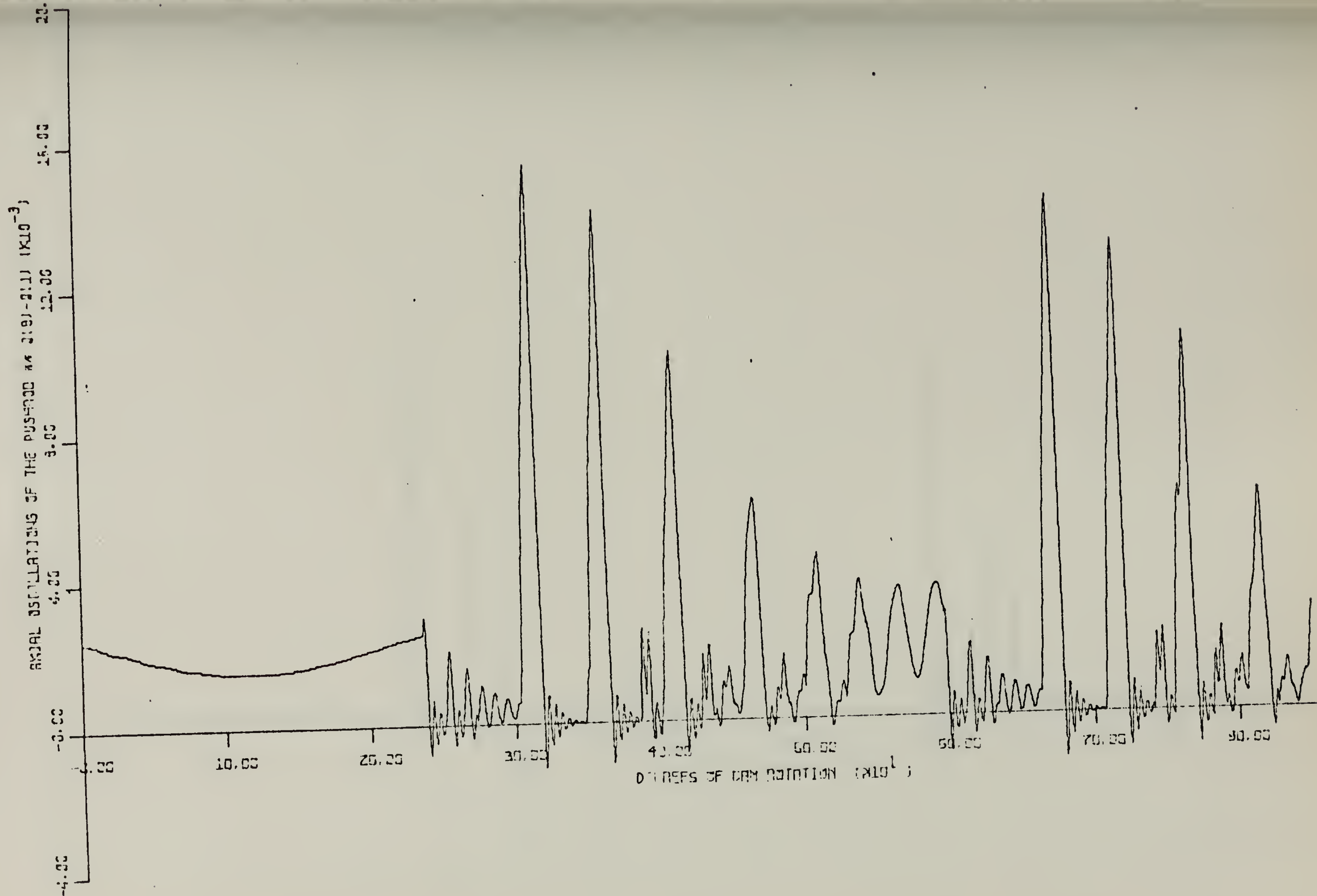


Figure F.6 THREE CLEARANCES 9,000 RPM VALVE SEAT

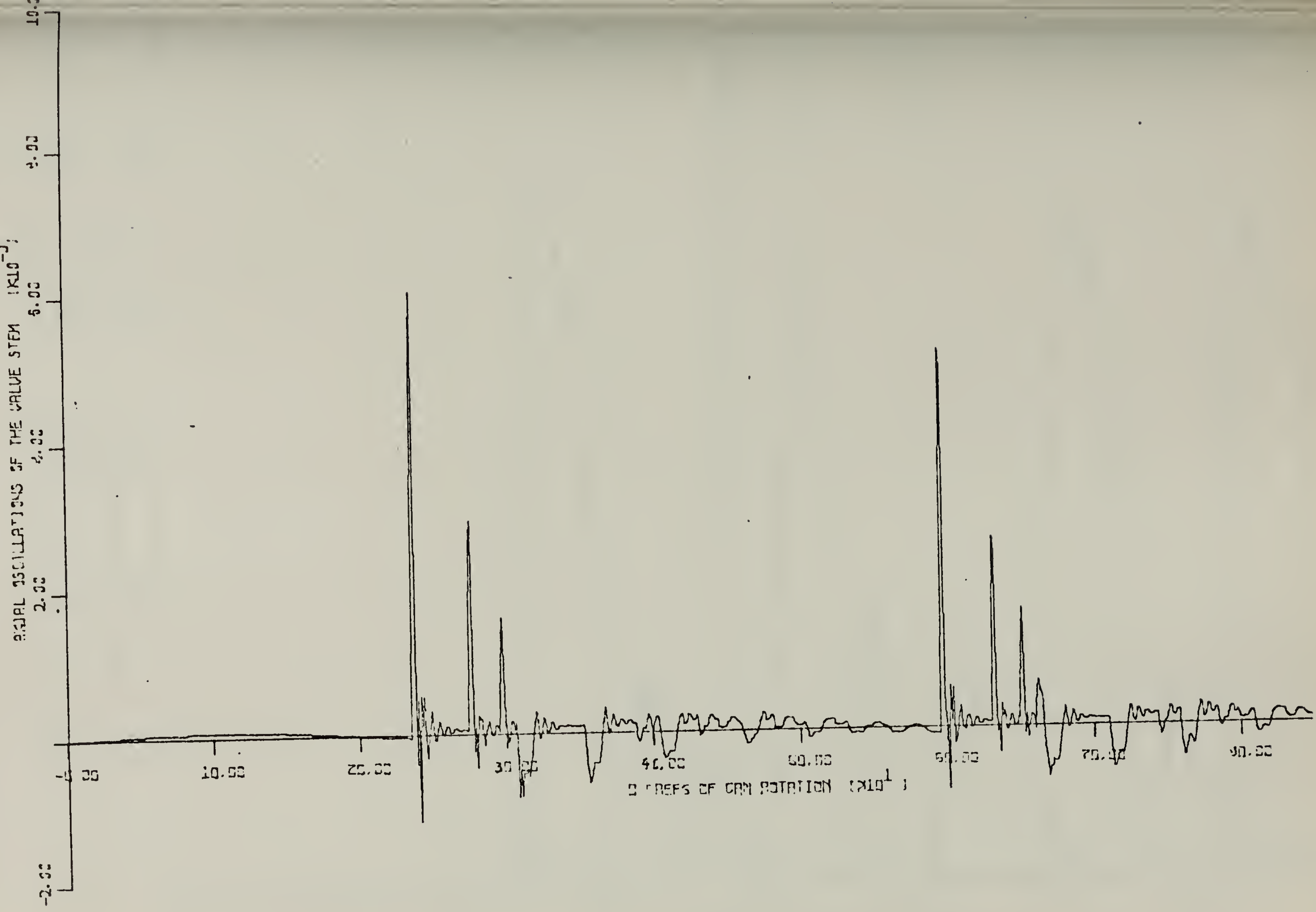


Figure F.7 THREE CLEARANCES 9,000 RPM VALVE SEAT

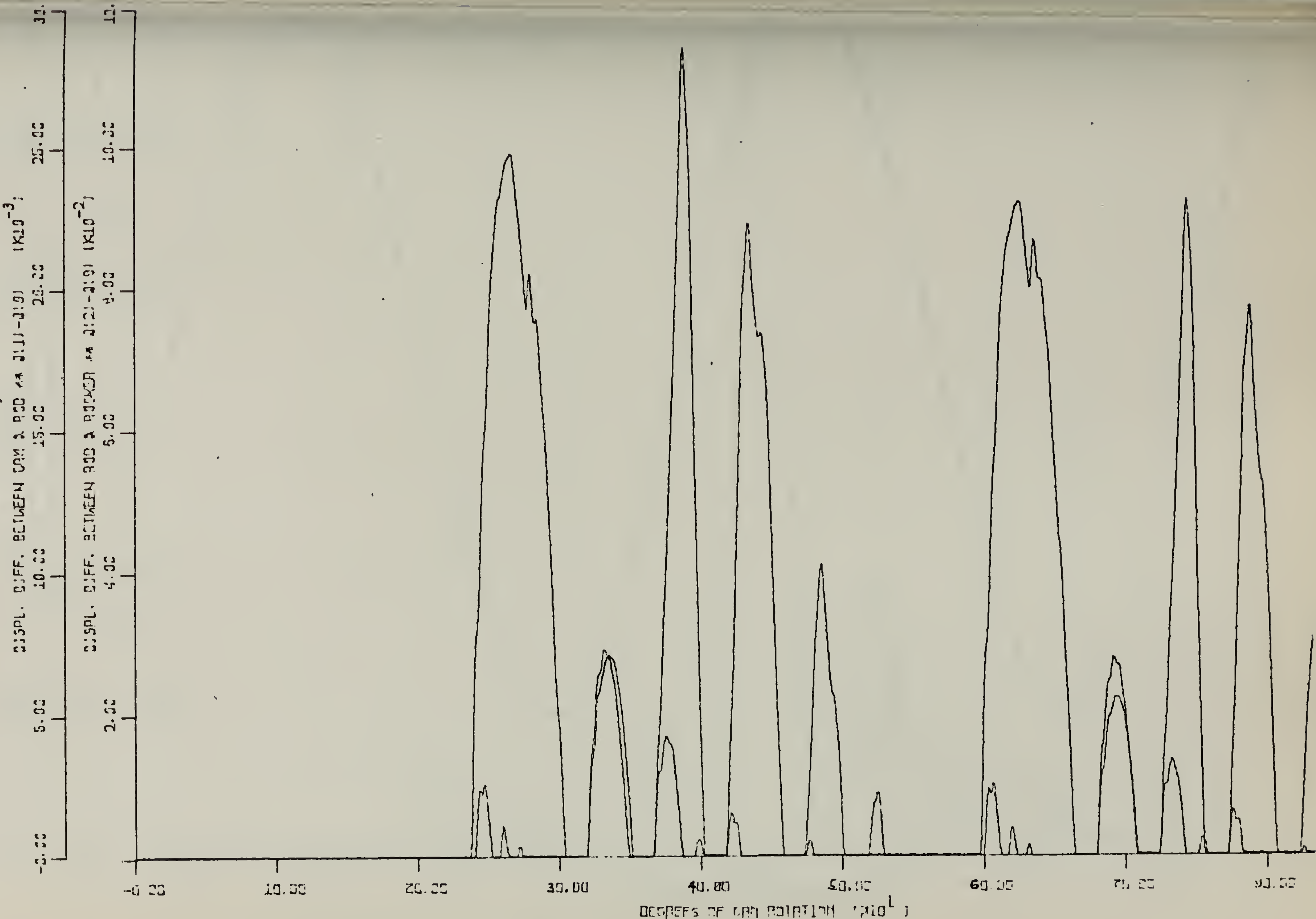


Figure F.8 THREE CLEARANCES 9,000 RPM VALVE SEAT

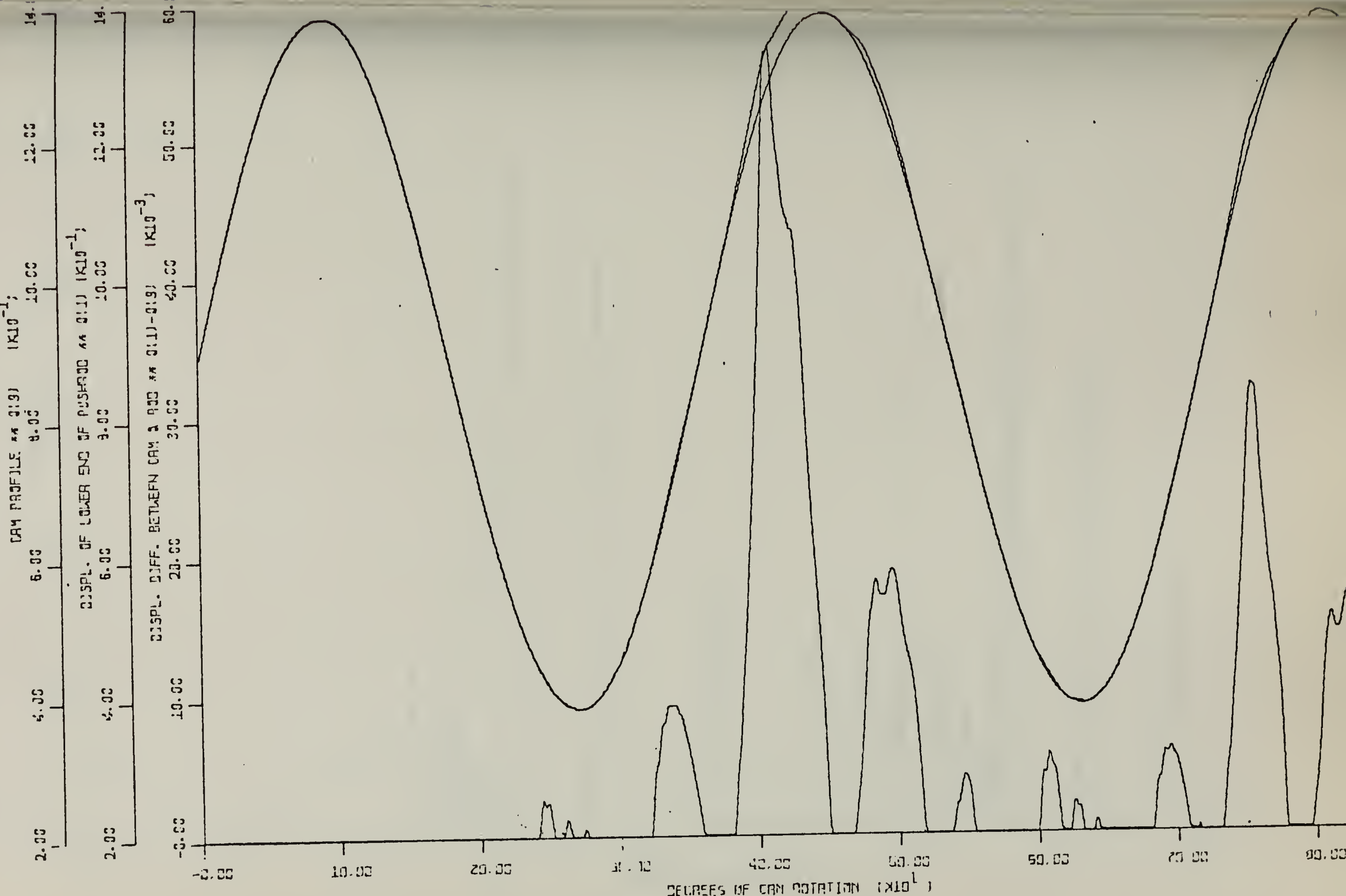


Figure F.9 THREE CLEARANCES 10,000 RPM VALVE SEAT

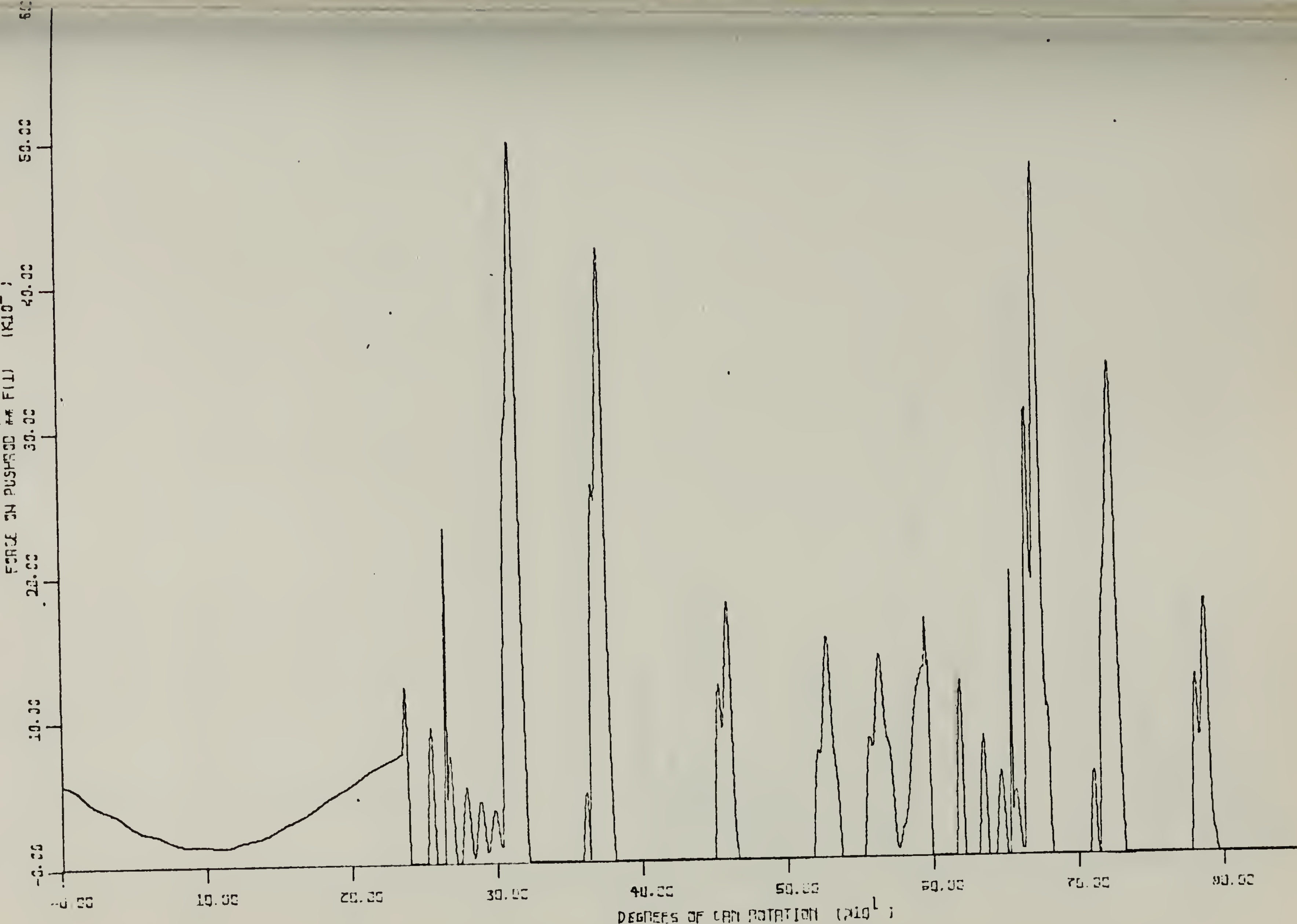


Figure F.10 THREE CLEARANCES 10,000 RPM VALUE SEAT

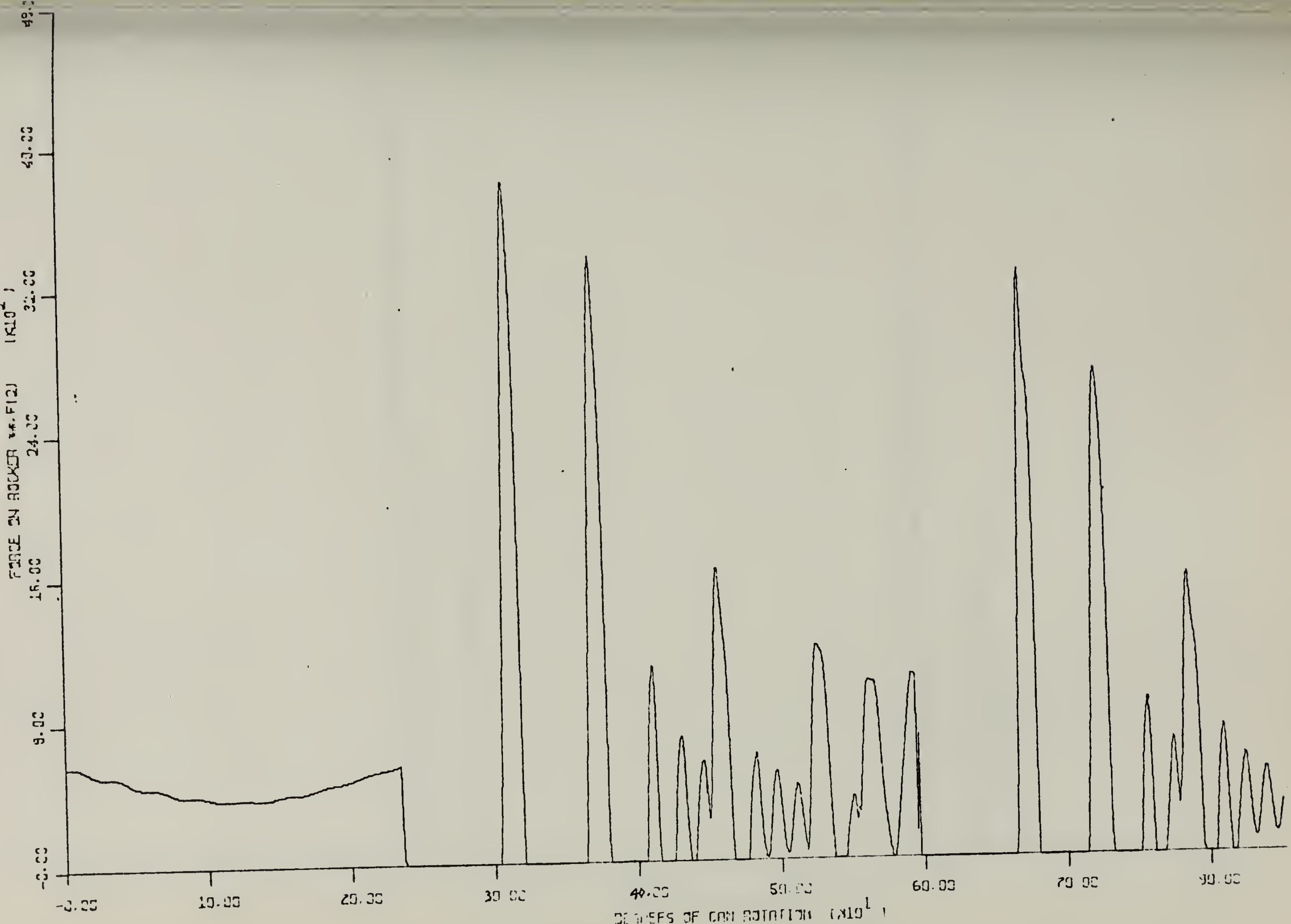


Figure F.11 THREE CLEARANCES 10,000 RPM VALVE SEAT

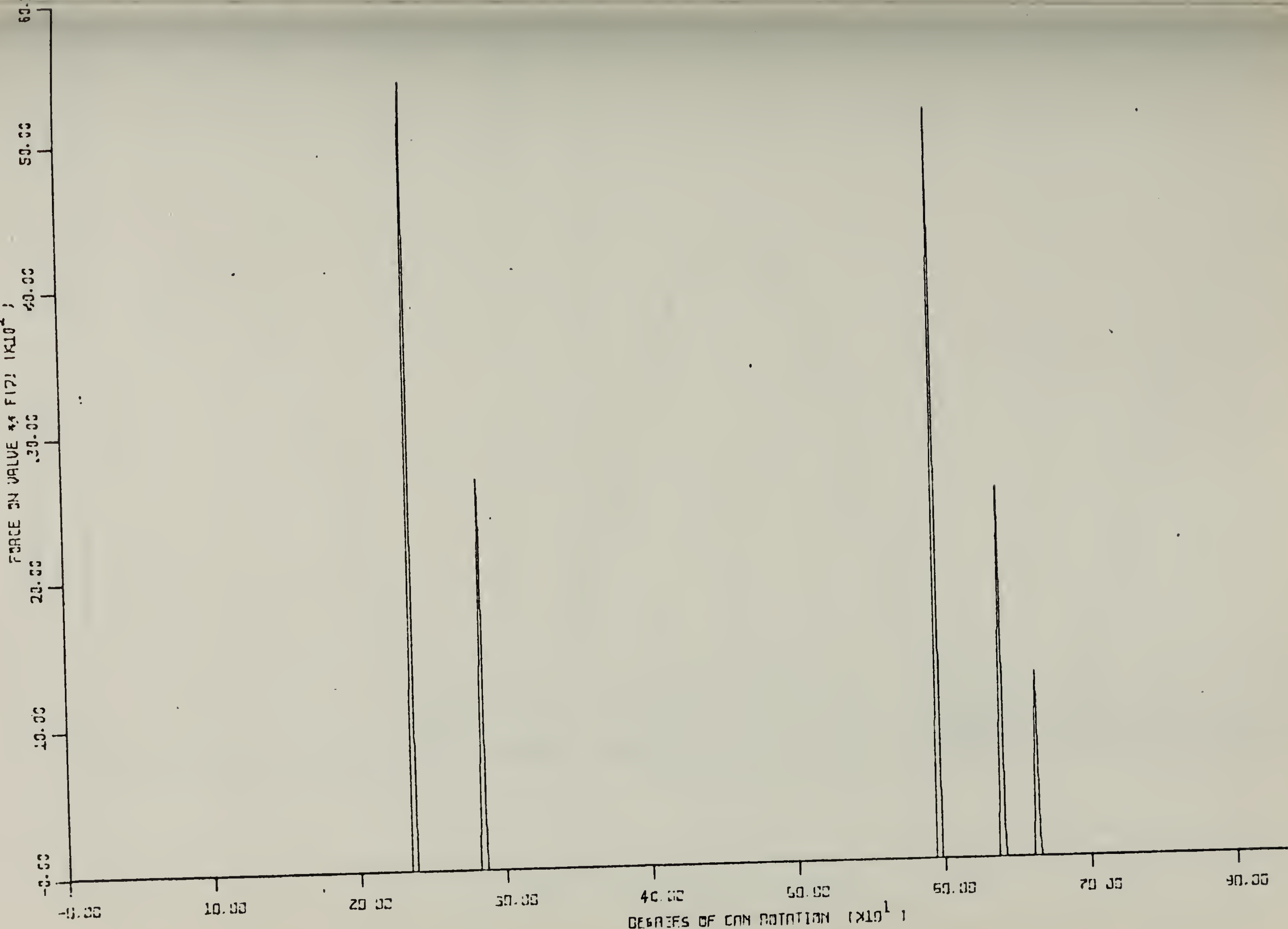


Figure F.12 THREE CLEARANCES 10,000 RPM VALVE SEAT

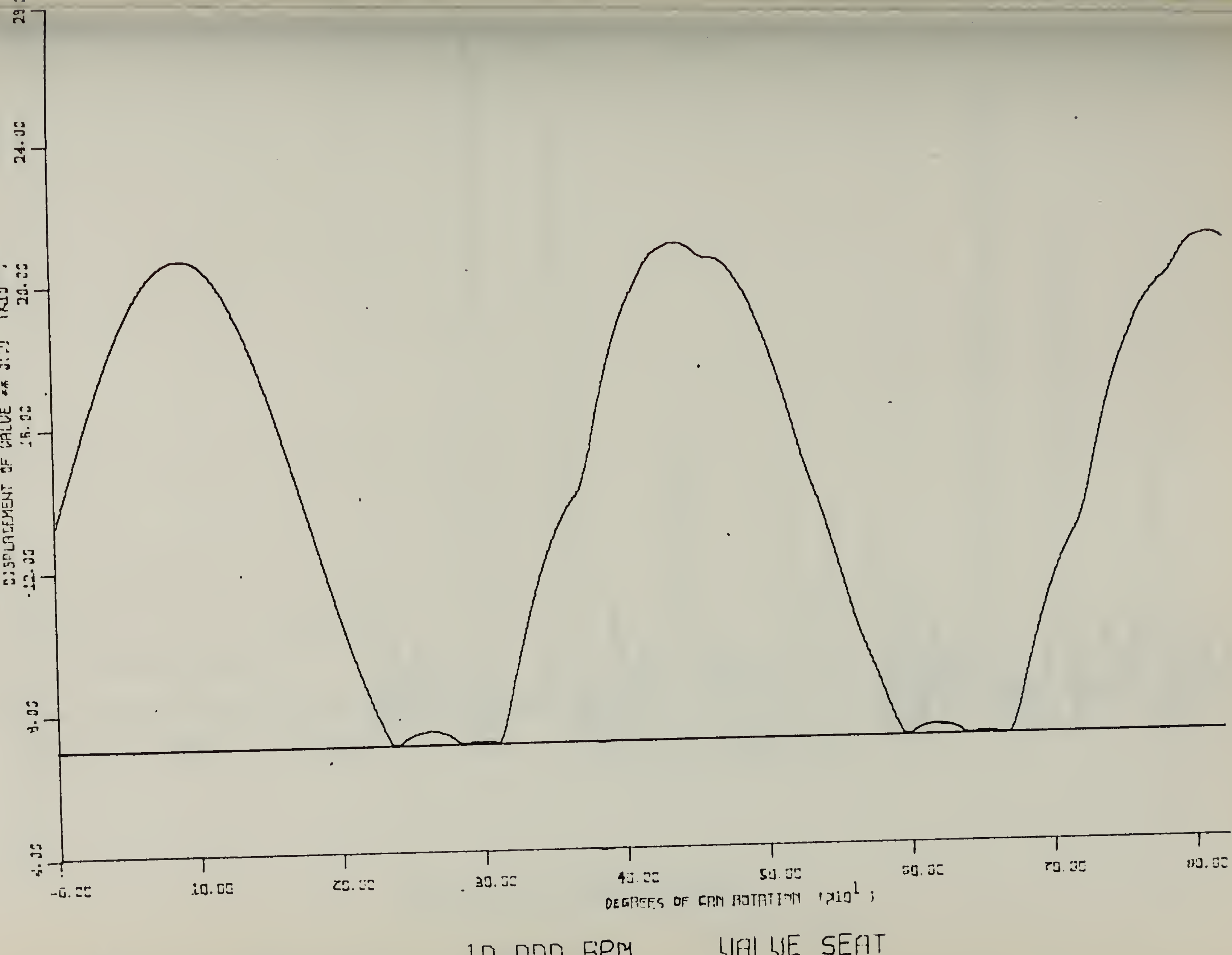


Figure F.13 THREE CLEARANCES

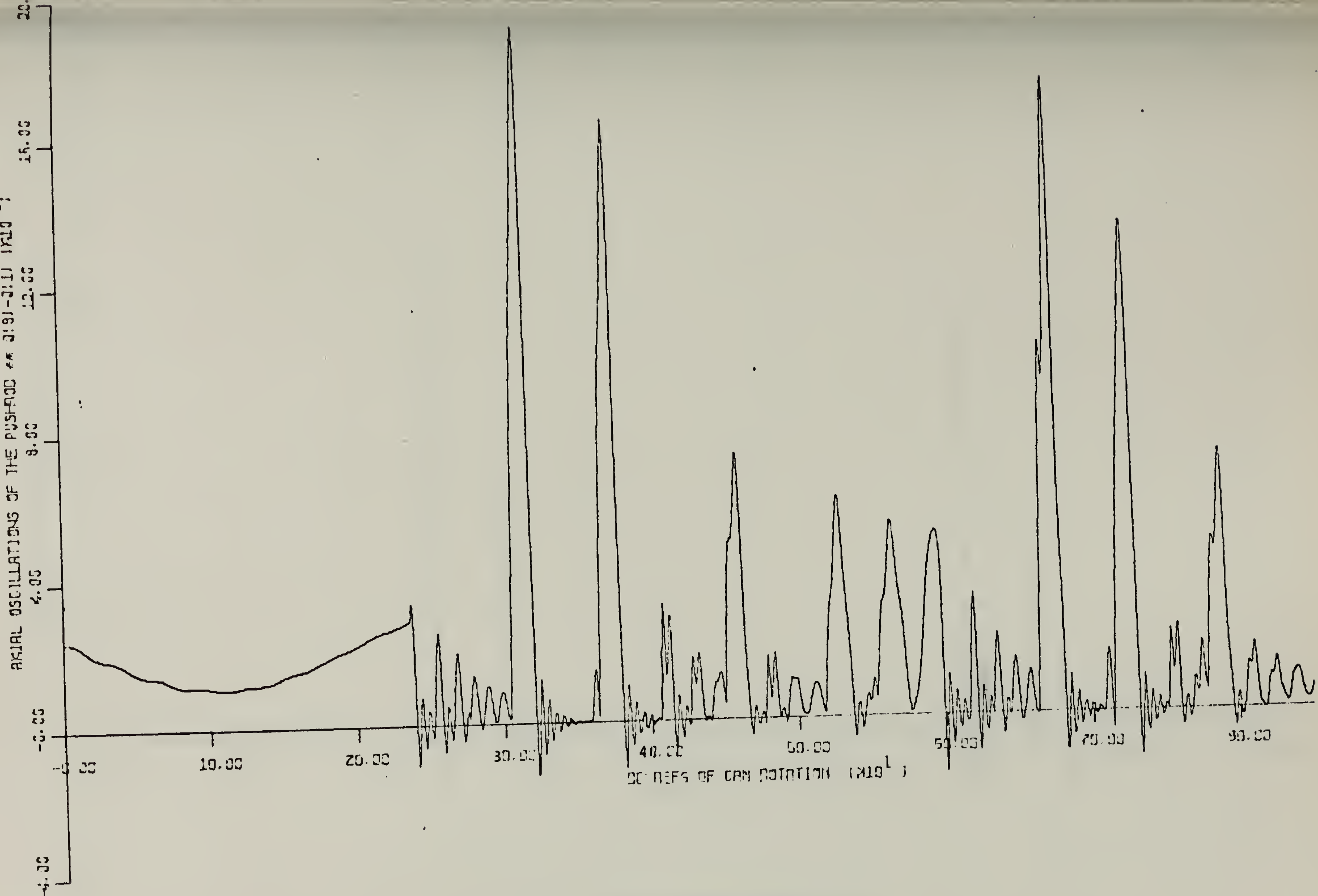


Figure F.14 THREE CLEARANCES

10,000 RPM VALVE SEAT

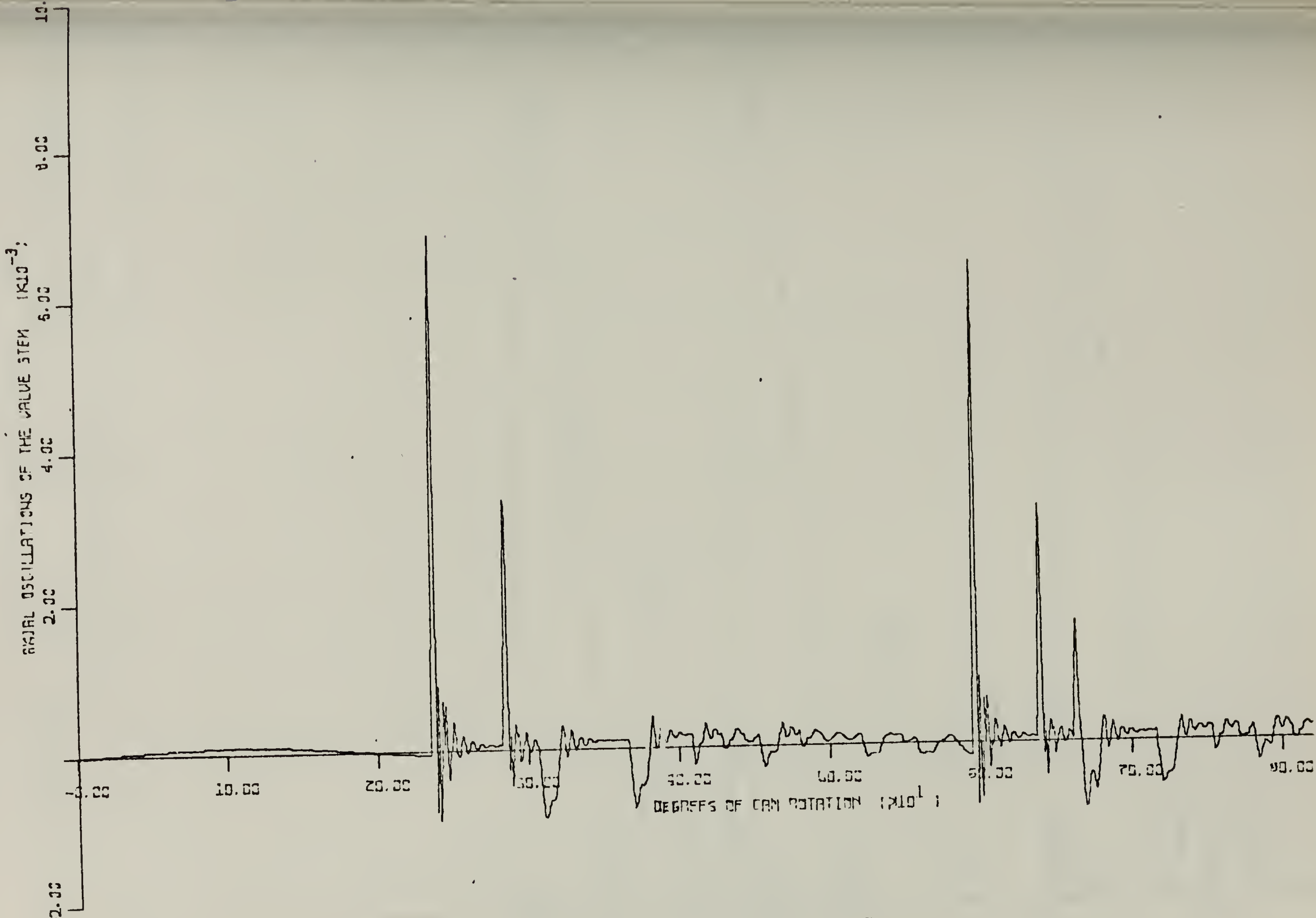


Figure F.15 THREE CLEARANCES

10,000 RPM VALVE SEAT

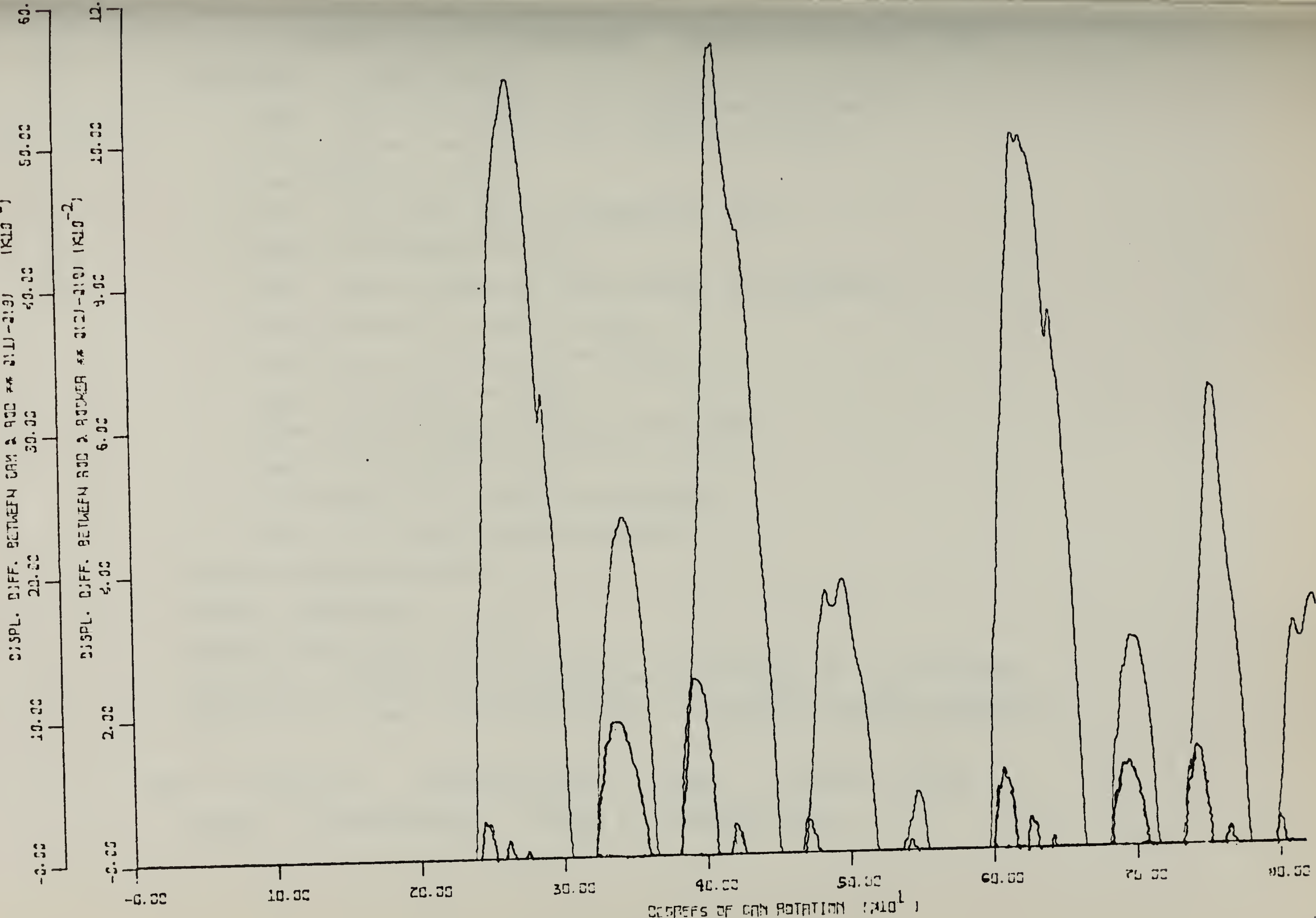


Figure F.16 THREE CLEARANCES 10,000 RPM VALVE SEAT

```
        GNILKA, C. W. 17 DECEMBER   THREE CLEARANCES  NO. 1
11,000 RPM     VALVE SEAT
     DISPL. DIFF. BETWEEN ROD & ROCKER ** Q(2)-Q(8)
     DISPL. DIFF. BETWEEN CAM & ROD ** Q(1)-Q(9)
     CAM PROFILE ** Q(9)
     DISPL. OF LOWER END OF PUSHROD ** Q(1)
     FORCE ON PUSHROD ** F(1)
     AXIAL OSCILLATIONS OF THE PUSHROD ** Q(8)-Q(1)
     DISPLACEMENT OF VALVE ** Q(7)
     POSITION OF VALVE SEAT ** Q(10)
     FORCE ON ROCKER ** F(2)
     AXIAL OSCILLATIONS OF THE VALVE STEM
     FORCE ON VALVE ** F(7)
     CAM PPOFILE AND DISPL. OF PUSHROD
     DISPL. OF VALVE AND VALVE SEAT
 NO AGT OUTPUT REQUESTED
 DAMPING REQUESTED
 CALCOMP GRAPH NOT DESIRED
 VALVE SEAT IMPOSED WITH AN INITIAL CLEARANCE OF  0.70 INCHES
 COORDINATES 2 AND 8 ARE NOT PINED - - THREE CLEARANCE PROBLEM

RPM = 11000.00    DAMPING COEF. = 1.000    CLEAR = 0.700
 XZERO=  8.8245022D-01    FZERO =  5.0185073D 02
```

IBM-360 COMPUTER PROGRAM LISTING

This listing is coded in FORTRAN IV, (Refs. 10-13), to run on an IBM-360/67, with OS/360 release 18. The simulation requires 234 K bytes of core storage for 850 degrees of cam rotation, including the CALCOMP graph buffer. The offline printer array (GRID) is not included in this core storage figure. Execution time is a function of the speed of rotation of the cam. For 11,000 RPM with 850 degrees, execution time is about 4.3 minutes. The primary output parameters fill arrays for SYSOUT = B and SYSOUT = C. The B queue is for transferring data, via punched cards, for use in the SDS-9300 computer. The C queue is for plotting graphs via the CALCOMP.

```
C
C-----------------------------------------------------------------------
C
C       CHARLES   W. GNILKA - - -  THESIS - - -  DECEMBER    1971
C     THE MODELING AND ANALYSIS OF AN ELASTIC MECHANISM WITH CLEARANCES
C            THREE CLEARANCES - - CAM -  VALVE SEAT  -  ROCKER
C
C-----------------------------------------------------------------------
C
      INTEGER*2 GRID(   1, 1)
      DIMENSION FPLOT( 850,11), Y1( 850,2), Y2( 850,2), Y3(850,2),
     1          Y4(850,2), DEG(1700), DEG1(850 ), DEG2( 850,2),
     2          YNAME(20,13), YNAME1(20,2), YNAME2(20,2)
      COMMON NUNIT, IPRINT
      EQUIVALENCE ( FPLOT(1,1),Y1(1,1) ), ( YNAME(1,1),YNAME1(1,1) ),
     1            ( FPLOT(1,2),Y2(1,1) ), ( YNAME(1,2),YNAME2(1,1) ),
     2            ( FPLOT(1,3),Y3(1,1) ), ( FPLOT(1,7),Y4(1,1) ),
     3            ( DEG(1)  ,DEG2(1,1) ), ( DEG(851)  ,DEG1(1) )
      DATA YNAME(19,1 )/'    '/, YNAME(20,1 )/'    '/
      DATA YNAME(19,2 )/'    '/, YNAME(20,2 )/'    '/
      DATA YNAME(19,3 )/'    '/, YNAME(20,3 )/'    '/
      DATA YNAME(19,4 )/'    '/, YNAME(20,4 )/'    '/
      DATA YNAME(19,5 )/'    '/, YNAME(20,5 )/'    '/
      DATA YNAME(19,6 )/'    '/, YNAME(20,6 )/'    '/
      DATA YNAME(19,7 )/'    '/, YNAME(20,7 )/'    '/
      DATA YNAME(19,8 )/'    '/, YNAME(20,8 )/'    '/
      DATA YNAME(19,9 )/'    '/, YNAME(20,9 )/'    '/
      DATA YNAME(19,10)/'    '/, YNAME(20,10)/'    '/
      DATA YNAME(19,11)/'    '/, YNAME(20,11)/'    '/
      DATA YNAME(19,12)/'    '/, YNAME(20,12)/'    '/
      DATA YNAME(19,13)/'    '/, YNAME(20,13)/'    '/
      IP = 0
      LINES = 80*IP + 1
      NTIME = 850
      IGU = YES
      IGU = NO
      SCALE =  1.1D0
      CALL THESIS(FPLOT, DEG, DEG1, DEG2, NTIME, LINES, IP, GRID, Y1,
     1            Y2, Y3, Y4, YNAME, YNAME1, YNAME2, IGU, SCALE)
      STOP
      END
```

```
      SUBROUTINE AGT(Q, QR4, WT, WTR4, XZERO, ROCKER, NPRINT, AGTLTH,
     2         REALTH, C, IUNIT, KAGT)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      REAL*4 QR4, WTR4 , RXZERO,RAMP,RRPM, DIFF
C
      INTEGER YES, C
C
      DIMENSION Q(1), QR4(1), AGTLTH(1), REALTH(1)
C
      COMMON NUNIT, IPRINT
      DATA YES/4HYES /, NO/4HNO  /
C
 5001 FORMAT(5E15.7)
 5002 FORMAT(5E15.7, I3)
C
C              *****************************************
C              **                                     **
C              **         OUTPUT PACKAGE FOR AGT      **
C              **           CONVERT TO REAL * 4       **
C              **                                     **
C              *****************************************
C
      IF(KAGT .NE. 1)  GO TO 10
      S = REALTH(2) / AGTLTH(2)
      THETA = DARSIN( XZERO/REALTH(2) )
      DIFF = DABS( Q(7) - Q(10) )
      WRITE(IUNIT,5001) DIFF
C
   10 QR4(1)  =     Q(1) - XZERO
      QR4(2)  =     Q(2) - XZERO
      QR4(8)  =     Q(8) - XZERO
      QR4(9)  =     Q(9) - XZERO
      QR4(4)  =     Q(4) + XZERO*ROCKER
      QR4(7)  =     Q(7) - XZERO*ROCKER
      QR4(3)  =   ( Q(3) + THETA )*S
      QR4(5)  =   ( Q(5) + THETA )*S
      QR4(6)  =   ( Q(6) + THETA )*S
      WTR4  =  WT
C
      WRITE(IUNIT,5001) (QR4(JA), JA  =  1,5)
      WRITE(7    ,5002) (QR4(JA), JA  =  6,9), OMEGA(I), CONFIG(I)
C
      RETURN
      END
```

```
      SUBROUTINE ARRAY(F, Q, QD, QDD, NDEG, FPLOT, NTIME)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      REAL*4 FPLOT
C
      INTEGER YES
C
      DIMENSION F(1), Q(1),QD(1),QDD(1),FPLOT(NTIME,1)
C
      COMMON NUNIT, IPRINT
C
      DATA YES/4HYES /, NO/4HNO  /
C
C
      FPLOT(NDEG,1 ) = Q(2) - Q(8)
      FPLOT(NDEG,2 ) = Q(1) - Q(9)
      FPLOT(NDEG,3 ) = Q(9)
      FPLOT(NDEG,4 ) = Q(1)
      FPLOT(NDEG,5 ) = F(1)
      FPLOT(NDEG,6 ) = Q(1) - Q(8)
      FPLOT(NDEG,7 ) = Q(7)
      FPLOT(NDEG,8 ) = Q(10)
      FPLOT(NDEG,9 ) = F(2)
      FPLOT(NDEG,10) = Q(7) + Q(4)
      FPLOT(NDEG,11) = F(7)
C
      RETURN
      END
```

```
      SUBROUTINE BKSTEP( BMIV, DA, DV, W, WT, SYSM, DAMP, STF, C, NPRINT
     1,                  KSYS, KPRINT, DP, QP, QDP, QDDP, ALFA, BETA,
     2                   CONFIG, FP, ICORD, NCLEAR, AMP, XZERO, DTPM, DT,
     3                   ISYS, NC, AMPW, W2, KSYSM1, KSYSM2, DTW, NSYS,
     4                   IDT, PIN2$8, IDIM, IQ, COLM, COLC, COLK, IDP,
     5                   QSAV, QDSAV, QDDSAV, FSAV, LAYER, ICHNG, IGU,
     6                   SCALE)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER CONFIG(16,1), YES, C, PIN2$8
C
      DIMENSION BMIV(KSYS,KSYS), DA(KSYS,KSYS), DV(KSYS,KSYS),
     1          SYSM(KSYS,KSYS,1), DAMP(KSYS,KSYS,1), STF(KSYS,KSYS,1),
     2          DP(1), FP(1), QP(1), QDP(1), QDDP(1), ALFA(1),
     3          BETA(1), ICORD(NCLEAR,1), IDIM(1), IQ(KSYS,1)
      DIMENSION QSAV(1), QDSAV(1), QDDSAV(1), FSAV(1), IDP(KSYS,1),
     1          COLM(KSYS,1), COLC(KSYS,1), COLK(KSYS,1), LAYER(1)
C
      COMMON NUNIT, IPRINT
C
      DATA YES/4HYES /, NO/4HNO  /
C
 5000 FORMAT(' ',10X,'FROM BKSTEP', 10X, 'PARTIAL DT =', 1PD15.7, 10X,
     1       ' PARTIAL WT =', D15.7)
 5001 FORMAT(' ',   5X, 'BACKSTEP REQUIRED ', I4, ' INTERATIONS TO ',
     1       'REACH A POSITIVE/ZERO FORCE BETWEEN COORD. ', I2, ' AND ',
     2       I2, '     (C=', I2, ')'   )
 5002 FORMAT('0', 10X, 'IN SUBROUTINE BKSTEP; NEGATIVE FORCE DETECTED ',
     1       '- - NEW C CHANGED TO', I2, ' FROM ', I2)
 5003 FORMAT(' ', 'FSAV' ,8(1PD15.7,1X) )
 5004 FORMAT(' ', '  QSAV',10(1PD12.4) )
 5005 FORMAT(' ', ' QDSAV',10(1PD12.4) )
 5006 FORMAT(' ', 'QDDSAV',10(1PD12.4) )
 5007 FORMAT(' ', 10X, ' DTPM = ', 1PD15.7, '  DTPMW =', D15.7, '  WT=',
     1       D15.7, '  DELWT=', D15.7, '  DELDT=', D15.7)
 5008 FORMAT(' ', 10X, ' CAMQ = ', 1PD15.7, '  CAMQD =', D15.7, '  CAMQD
     1D='    D15.7)
 5009 FORMAT(' ', 10X, ' DTPM = ', 1PD15.7, '       C =', I2)
 5010 FORMAT(' ',          'C=', I2,   ' FORCE =', 1P8D15.7)
 5011 FORMAT(' ',  3X,     '    FORCE =', 1P8D15.7)
 5012 FORMAT(' ',    4X, 'BACKSTEP REQUIRED ', I4, ' INTERATIONS TO ',
     1       'REACH A POSITIVE/ZERO DISPLACEMENT BETWEEN COORD. ',I2,
     2       ' AND ', I2, '      (C=', I2, ')'   )
 5013 FORMAT(' ',  '      DISPL DIFF. AT COORD 1 =', 1PD15.7,
     1             '      DISPL DIFF. AT COORD 2 =', 1PD15.7,
```

```
     2          '      DISPL DIFF. AT COORD 7 =', 1PD15.7)
 5014 FORMAT(' ',  'DEGREES=',F9.3, '  CSAVE=',I2, '   C=',I2,
     1       '  DISPL 1,2,7=', 1P3D11.3, '  FORCES 1,2,7 =',3D11.3)
 5017 FORMAT('0', 10X, ' VALUES CALCULATED AFTER BACKSTEP TAKEN:')
C
      EPS1 = 1.0D-1
      EPS2 = 1.0D-6
C
      PI = 3.14159265359ODO
      DEGOPI = 180.0DO/PI
C
      IK = 0
      DTPMW = DTPM*W
      DELWT = DTPMW/ 500.0DO
      WT = WT + DTPMW - DTW
      WTSAV = WT
      DELDT = DELWT/W
C
      IF(NPRINT .EQ. YES)
     1        WRITE(6,5007)  DTPM, DTPMW, WT, DELWT, DELDT
C
C             ****************************************************
C             **                                                **
C             **          SAVE   OLD   PARAMETERS               **
C             **                                                **
C             ****************************************************
C
    4 DO 5 JA = 1,KSYS
      FP(JA) = FSAV(JA)
    5 CONTINUE
C
      DO  15 JA = 1,NSYS
       CALL ADJUST(JA,JA,QP,QDP,QDDP,QSAV,QDSAV,QDDSAV)
   15 CONTINUE
C
      IF(NPRINT .EQ. NO)  GO TO 20
      WRITE(6,5003)  (  FSAV(JA), JA = 1, KSYS)
      WRITE(6,5004)  (  QSAV(JA), JA = 1, NSYS)
      WRITE(6,5005)  ( QDSAV(JA), JA = 1, NSYS)
      WRITE(6,5006)  (QDDSAV(JA), JA = 1, NSYS)
C
   20 IF(NPRINT .EQ. YES) WRITE(NUNIT,5000) DTPM, WT
      IK = IK + 1
C
      CALL CAM(WT, AMP, AMPW, XZERO, W2, QP, QDP, QDDP)
C
      IF(NPRINT .EQ. YES)
     1        WRITE(6,5008)  QP(9), QDP(9), QDDP(9)
```

```
C
      CALL DEL( DTPM, DTO2, DT203, DT206, BMIV, DA, DV, SYSM(1,1,C),
     1         DAMP(1,1,C), STF(1,1,C), ISYS, C, NPRINT, KSYS, KPRINT)
C
      IF(NPRINT .EQ. YES)
     1       WRITE(6,5009)  DTPM, C
C
      CALL BRANCH(C, FP, KSYS, ISYS, QP, QDP, QDDP, QDP, ICORD, DT,
     1            NPRINT, NCLEAR, 0, PIN2$8, IDIM)
C
      IF(NPRINT .EQ. YES)
     1       WRITE(6,5010)  C,  (FP(JA), JA= 1,KSYS)
C
      CALL NEWMRK( ISYS, KSYS, C, STF, DAMP, SYSM, DP, DTPM, DTO2,
     1            DT206, DT203, QP, QDP, QDDP, BMIV, DA, DV, NC, ALFA,
     2            BETA, CONFIG, FP, ICORD, NCLEAR, NPRINT, IQ, COLM,
     3         COLC, COLK, IDP, PIN2$8, IDIM, LAYER, IGU, QDDSAV,
     4           QDSAV, QSAV, 0.0, 0)
      IF(NPRINT .EQ. YES)
     1       WRITE(6,5011)       (FP(JA), JA= 1,KSYS)
C
      DISPL1 = QP(1) - QP(9)
      DISPL2 = QP(2) - QP(8)
      DISPL7 = QP(7) - QP(10)
C
      IF(NPRINT .EQ. YES)
     1       WRITE(NUNIT,5013) DISPL1,  DISPL2, DISPL7
C
      INDEX = ICORD(IDT,1)
      NDEX = ICORD(IDT,2)
      IF(CONFIG(C,IDT) .EQ. 0)  GO TO 25
C
      IF(FP(INDEX) .GE. 0.0  )  GO TO 24
      GO TO 30
C
   24 IF(KPRINT .EQ. YES  .AND.  IK .GT. 1)
     1     WRITE(NUNIT,5001) IK, INDEX, NDEX, C
      GO TO 40
C
C
```

```
   25 DISPL = QP(INDEX) - QP(NDEX)
C
      IF(DISPL .GE. 0.0)  GO TO 28
      IF( DABS(DISPL) .LE. EPS2)    GO TO 28
      GO TO 30
C
   28 IF(KPRINT .EQ. YES   .AND.   IK .GT. 1)
     1     WRITE(NUNIT,5012) IK, INDEX, NDEX, C
      GO TO 40
C
C
C
   30 DTPM = DTPM - DELDT
      WT = WT - DELWT
C
      IF(WT .LE. WTSAV)  RETURN
C
      GO TO 4
C
   40 CONTINUE
C
      DEGR = WT*DEGOPI
      NDEG = DEGR
C
      WRITE(NUNIT,5014) DEGR,C,ICHNG,DISPL1,DISPL2,DISPL7,
     1                  FP(1), FP(2), FP(7)
      IF(NPRINT .EQ. NO)  RETURN
C
C
      WRITE(NUNIT,5017)
C
      CALL DEBUG( 2, Q, QD, QDD, F, KSYS, ISYS, DP, NDEG, NPRINT, DEGR,
     1            C)
C
      RETURN
      END
```

```
      SUBROUTINE BRANCH(C,F,KSYS,ISYS,Q,QD,QDD,QDSAV,ICORD,DT,
     1                  NPRINT, NCLEAR, IMP, PIN2$8, IDIM)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER YES, C, PIN2$8
C
      DIMENSION Q(1), QD(1), QDD(1),QDSAV(1), F(1), ICORD(NCLEAR,1),
     1           IDIM(1)
C
      COMMON NUNIT, IPRINT
C
      DATA YES/4HYES /, NO/4HNO  /
C
    5 ISYS = IDIM(C)
C
      GO TO (  10, 20,  30, 40, 50, 60, 70, 80), C
C
C     CONFIGURATION - I - NO CONTACT
   10 F(1) = 0.0D0
      F(2) = 0.0D0
      F(7) = 0.0D0
      GO TO 100
C
C
C     CONFIGURATION -II - CONTACT AT VALVE SEAT
   20 CONTINUE
      CALL ADJUST(7,10, Q, QD, QDD, Q, QD, QDD)
C
      F(1) = 0.0D0
      F(2) = 0.0D0
      GO TO 100
C
C
C     CONFIGURATION - III - CONTACT AT ROCKER
   30 F(1) = 0.0D0
      F(7) = 0.0D0
      GO TO 100
C
C
C     CONFIGURATION -  IV - CONTACT AT ROCKER AND VALVE SEAT
   40 CONTINUE
C
      CALL ADJUST(7,10, Q, QD, QDD, Q, QD, QDD)
C
      F(1) = 0.0D0
```

```
      GO TO 100
C
C
C
C     CONFIGURATION - V -CONTACT AT CAM
   50 CONTINUE
C
      CALL ADJUST(1,9, Q, QD, QDD, Q, QD, QDD)
C
      F(2) = 0.0D0
      F(7) = 0.0D0
      GO TO 100
C
C
C
C     CONFIGURATION -VI -CONTACT AT CAM AND VALVE SEAT
   60 CONTINUE
C
      CALL ADJUST(1,9, Q, QD, QDD, Q, QD, QDD)
      CALL ADJUST(7,10, Q, QD, QDD, Q, QD, QDD)
C
      F(2) = 0.0D0
      GO TO 100
C
C
C
C     CONFIGURATION - VII - CONTACT AT ROCKER AND CAM
   70 CONTINUE
C
      CALL ADJUST(1,9, Q, QD, QDD, Q, QD, QDD)
C
      F(7) = 0.0D0
      GO TO 100
C
C
C
C     CONFIGURATION - VIII - CONTACT AT ROCKER, CAM, AND VALVE SEAT
   80 CONTINUE
C
      CALL ADJUST(1,9, Q, QD, QDD, Q, QD, QDD)
      CALL ADJUST(7,10, Q, QD, QDD, Q, QD, QDD)
C
  100 CONTINUE
      IF(IMP .NE. 0)  QDD(IMP) = ( QD(IMP)-QDSAV(IMP) ) / DT
C
      RETURN
      END
```

```
      SUBROUTINE CONMAT(NC, CONFIG, IDP, NPRINT, KSYS)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER CONFIG(16,4), CSAVE, TWON, TWONP1, YES
C
      DIMENSION IDP(KSYS,1)
C
      COMMON NUNIT, IPRINT
C
      DATA  YES/4HYES /, NO/4HNO  /
C
C
C                ********************************************
C                **                                        **
C                **      ESTABLISH CONFIGURATION           **
C                **                MATRIX                  **
C                **                                        **
C                ********************************************
C
  100 NC = 4
      TWON = 2**NC
      TWONP1 = TWON + 1
      NCM1 = NC - 1
      NCM2 = NC - 2
      NCM3 = NC - 3
C
      DO  110  I = 2,TWON,2
      CONFIG(I,NC) = 1
  110 CONTINUE
C
      DO  120  I = 2,TWON,4
      CONFIG(I+1,NCM1) = 1
      CONFIG(I+2,NCM1) = 1
  120 CONTINUE
C
      DO 130 I = 1,NC
      CONFIG(I+4,NCM2) = 1
      CONFIG(I+12,NCM2) = 1
  130 CONTINUE
```

```
C
C          ********************************************
C          **                                        **
C          **              ESTABLISH                 **
C          **            MATRIX   IDP                **
C          **                                        **
C          ********************************************
C
      DO  140  JA = 1,KSYS
C
      IDP(JA,1) = CONFIG(JA,2)
      IDP(JA,2) = 0
      IDP(JA,3) = CONFIG(JA,4) * 7
C
  140 CONTINUE
C
      NC = 8
C
      RETURN
      END
```

```
      SUBROUTINE DEBUG(IBLOCK, Q, QD, QDD, F, KSYS, ISYS, DP, NDEG,
     1                 NPRINT, DEGR, C)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER YES, C
C
      DIMENSION Q(10), QD(10), QDD(10), F(10), DP(8)
C
      COMMON NUNIT, IPRINT
C
      DATA YES/4HYES /, NO/4HNO  /
C
 5001 FORMAT(' ','****************************************************',
     1'***************************************************************')
 5002 FORMAT(' ','Q    ',1P7D16.7, D15.7)
 5003 FORMAT(' ','QD   ',1P7D16.7, D15.7)
 5004 FORMAT(' ','QDD  ',1P7D16.7, D15.7)
 5005 FORMAT(' ','DEL P',1P7D16.7, D15.7)
 5007 FORMAT(' ',  '      DISPL DIFF. AT COORD 1 =', 1PD15.7,
     1              '      DISPL DIFF. AT COORD 2 =', 1PD15.7,
     2              '      DISPL DIFF. AT COORD 7 =', 1PD15.7)
 5037 FORMAT(' ','FORCE',1P7D16.7, D15.7)
 5061 FORMAT(' ', 16X,I5,'  DEGREES =',  F9.3, 15X, 'CONFIGURATION ',I2)
 5063 FORMAT(' ','CAMQ=',1PD16.7, 11X,'CAMQD =', D16.8, 10X,'CAMQDD =',
     1    D16.8)
C
      DISPL1 = Q(1) - Q(9)
      DISPL2 = Q(2) - Q(8)
      DISPL7 = Q(7) - Q(10)
      GO TO(1,2,3,4,5), IBLOCK
C
    1 CONTINUE
      WRITE(NUNIT,5061) NDEG,  DEGR, C
      WRITE(NUNIT,5007) DISPL1,  DISPL2,  DISPL7
      WRITE(NUNIT,5063) Q(9), QD(9), QDD(9)
      WRITE(NUNIT,5002) (   Q(JA), JA = 1, KSYS )
      WRITE(NUNIT,5003) (  QD(JA), JA = 1, KSYS )
      WRITE(NUNIT,5004) (QDD(JA), JA = 1, KSYS )
      WRITE(NUNIT,5037) (   F(JA), JA = 1, KSYS )
      IF(NPRINT .EQ. YES)  WRITE(NUNIT,5001)
C
      RETURN
C
C
C
```

```
    2 CONTINUE
      IF(NDEG .GT. 200)  NPRINT = YES
      WRITE(NUNIT,5061) NDEG,  DEGR, C
      WRITE(NUNIT,5007) DISPL1,  DISPL2,  DISPL7
      WRITE(NUNIT,5063) Q(9), QD(9), QDD(9)
      WRITE(NUNIT,5002) (  Q(JA), JA = 1, KSYS )
      WRITE(NUNIT,5003) ( QD(JA), JA = 1, KSYS )
      WRITE(NUNIT,5004) (QDD(JA), JA = 1, KSYS )
      WRITE(NUNIT,5037) (  F(JA), JA = 1, KSYS )
C
      RETURN
C
C
    3 CONTINUE
C
      RETURN
C
C
    4 CONTINUE
C
      RETURN
C
C
    5 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE DEL( DT, DT02, DT203, DT206, BMIV, DA, DV, EM,
     1 SE, CA, KS, C, NPRINT, KSYS, KPRINT)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER YES, C
C
      DIMENSION BMIV( KSYS,KSYS), DA( KSYS,KSYS), DV( KSYS,KSYS)
      DIMENSION   EM( KSYS,KSYS), SE( KSYS,KSYS), CA( KSYS,KSYS)
C
      COMMON NUNIT, IPRINT
      DATA YES/4HYES /, NO/4HNO  /
C
 1000 FORMAT('0',5X,'DT =' 1PD12.5,'  DT/2 =',D12.3,
     1'  DT**2/3 =',D12.3,'  DT**2/6 =',D12.3)
C
      DT02  = DT*0.5D0
      DT203 = DT**2/3.0D0
      DT206 = DT203*0.5D0
C
      IF(NPRINT .EQ. NO) GO TO 10
      WRITE(NUNIT,1000) DT, DT02, DT203, DT206
C
   10 DO 20  JA = 1, KS
C
      DO 20  JB = 1, KS
      BMIV(JA,JB) = EM(JA,JB) + DT02*SE(JA,JB) + DT206*CA(JA,JB)
C
   20 CONTINUE
C
      CALL DINVS( BMIV, KS,KSYS)
C
      DO 30  JA = 1, KS
C
      DO 30  JB = 1, KS
      DA( JA, JB) = 0.0D0
      DV( JA, JB) = 0.0D0
C
      DO 30  JC = 1, KS
      DA( JA, JB) = DA( JA, JB) + BMIV( JA, JC)*SE( JC, JB)
      DV( JA, JB) = DV( JA, JB) + BMIV( JA, JC)*CA( JC, JB)
C
   30 CONTINUE
C
      RETURN
      END
```

```
C
C------------------------------------------------------------------------
C
C                 SUBROUTINE CODED BY DR. R. C. WINFREY
C                     MODIFIED BY C. W. GNILKA
C
C------------------------------------------------------------------------
      SUBROUTINE  DINVS(A,NMAX,KSYS)
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER YES
C
      DIMENSION A(KSYS,KSYS)
C
      COMMON NUNIT, IPRINT
C
      DATA  YES/4HYES /, NO/4HNO  /
C
      DO 200 N=1,NMAX
      D=A(N,N)
C
      IF(D.EQ.0.) GO TO 300
C
      DO 100 J=1,NMAX
  100 A(N,J)=-A(N,J)/D
C
      DO 150 I=1,NMAX
      IF(N-I) 110,150,110
C
  110 DO 140 J=1,NMAX
      IF(N-J) 120,140,120
C
  120 A(I,J)=A(I,J)+A(I,N)*A(N,J)
  140 CONTINUE
C
  150 A(I,N)=A(I,N)/D
      A(N,N)=1.0/D
  200 CONTINUE
C
      RETURN
  300 WRITE(6,320)  N,D
  320 FORMAT('0',//,10X,'ERROR RETURN FROM DINVS.  MATRIX SINGULAR. ',
     1  'DIAGONL ELEMENT' ,I4,3H = ,1PD20.12)
C
      RETURN
      END
```

```
C
C-----------------------------------------------------------------------
C
C      THE FOLLOWING PLOT SUBROUTINES CODED BY DR. R. C. WINFREY
C
C-----------------------------------------------------------------------
C


      SUBROUTINE DRUP(X,Y,K,N,XSIZE,YSIZE,NPT,LABEL,TITLE,XNAME,YNAME)
      DIMENSION TITLE(1),XNAME(1),YNAME(20,1),LABEL(1)
      DIMENSION X(1),Y(N,1)
      CALL      DRUBL(X,Y,K,N,XSIZE,YSIZE,NPT,LABEL,TITLE,XNAME,YNAME)
C
      RETURN
      END
C
C
C
C
C


      SUBROUTINE DRUPB(X,Y,K,N,XSIZE,YSIZE,NPT,LABEL,TITLE,XNAME,YNAME)
      DIMENSION TITLE(1),XNAME(1),YNAME(20,1),LABEL(1)
      DIMENSION X(1),Y(N,1)
      CALL      DRUB(X,Y,K,N,XSIZE,YSIZE,NPT,LABEL,TITLE,XNAME,YNAME)
C
      RETURN
      END
C
C
C
C
C


      SUBROUTINE DRUBL(X,Y,K,N,XSIZE,YSIZE,NPT,LABEL,TITLE,XNAME,YNAME)
      DIMENSION TITLE(1),XNAME(1),YNAME(20,1),LABEL(1)
      DIMENSION X(1),Y(N,1),YMIN(20),DY(20)
      PAPER=11.
      CALL PLOTS
      CALL PLOT(-11.,0.,-3)
      CALL SYMBOL(0.,0.,.14,LABEL,0.,80)
      CALL PLOT(0.,4.,-3)
      NSIGN=1
      IF(NPT.LT.0) NSIGN=-1
      NPA=NPT*NSIGN
```

```
      XST=FLOAT(K)*.65
      XSTP=XST+(PAPER-XSIZE)*.5
      IF(XSIZE.GT.8.01) XSTP=XST+1.
      XAXIS=XSIZE-XST
      CALL SCALE(X,N,1,XAXIS,0.,XMIN,DX)
      DO 100 JA=1,K
      CALL SCALE(Y(1,JA),N,1,YSIZE,0.,YMIN(JA),DY(JA))
  100 CONTINUE
      DO 130 JA=1,N
  130 X(JA)=X(JA)+XSTP
      YMAX=YMIN(1)+DY(1)*YSIZE
      YS=ABS(YMIN(1))/DY(1)
      IF(YMAX.LT.0.) YS=0.
      IF(YMIN(1). GT.0.) YS=0.
      IF(XSIZE.GT.8.01) GO TO 200
      CALL CENT(XNAME,NTT)
      NLT=-NTT
      CALL EXISA(XSTP,YS,XNAME,NLT,XAXIS,0.,XMIN,DX)
      CALL EXISA(XSTP,YS,XNAME,NLT,XAXIS,0.,XMIN,DX)
      JB=1
      DO 160 JA=1,K
      JB=JB+1
      IF(JB.EQ.8) JB=2
      YST=XSTP-.65*FLOAT(JA-1)
      CALL CENT(YNAME(1,JA),NTT)
      CALL EXISA(YST,0.,YNAME(1,JA),NTT,YSIZE,90.,YMIN(JA),DY(JA))
      JC=NPT
      IF(NPA.EQ.10) JC=JB*NSIGN
      CALL LIND(X,Y(1,JA),N,1,JC)
      CALL LIND(X,Y(1,JA),N,1,JC)
      CALL LIND(X,Y(1,JA),N,1,JC)
  160 CONTINUE
      YL=-.8
      CALL CENT(TITLE,NTT)
      XL=PAPER*.5-FLOAT(NTT/2)*.18
      CALL SYMBOL(XL,  YL,.21,TITLE,0.,NTT)
      YAX=YSIZE+3.
      GO TO 300
  200 XFAR=(PAPER+YSIZE)*.5
      YS=XFAR-YS
      CALL CENT(XNAME,NTT)
      NLT=-NTT
      CALL EXISA(YS,XSTP,XNAME,NLT,XAXIS,90.,XMIN,DX)
      JB=1
      DO 260 JA=1,K
      JB=JB+1
      IF(JB.EQ.8) JB=2
      YST=XSTP-.65*FLOAT(JA-1)
```

```
      CALL CENT(YNAME(1,JA),NTT)
      CALL EXISA(XFAR,YST,YNAME(1,JA),NTT,YSIZE,180.,YMIN(JA),DY(JA))
      DO 230 JC=1,N
  230 Y(JC,JA)=XFAR-Y(JC,JA)
      JC=NPT
      IF(NPA.EQ.10) JC=JB*NSIGN
      CALL LIND(Y(1,JA),X,N,1,JC)
      CALL LIND(Y(1,JA),X,N,1,JC)
      CALL LIND(Y(1,JA),X,N,1,JC)
      DO 260 JC=1,N
      Y(JC,JA)=XFAR-Y(JC,JA)
  260 CONTINUE
      XL=XFAR+.8
      IF(YS.LT..01) XL=XL+.55
      CALL CENT(TITLE,NTT)
      YL=XSIZE*.5+1.-FLOAT(NTT/2)*.18
      CALL SYMBOL(XL,YL,.21,TITLE,90.,NTT)
      YAX=XSIZE+3.
  300 CALL PLOT(0.,YAX,-3)
      CALL SYMBOL(0.,0.,.14,LABEL,0.,80)
      CALL PLOT(1.25,4.,-3)
      CALL PLOTE
      WRITE(6,400)(TITLE(J),J=1,20)
  400 FORMAT(/////20X,75HA PLOT HAS BEEN DRAWN BY SUBROUTINE DRUP CONTAI
     1NING THE FOLLOWING TITLE -      ,////20X,20A4,////)
C    RESTORE ORIGINAL VALUES OF X AND Y
      DO 420 JA=1,N
      X(JA)=(X(JA)-XSTP)*DX+XMIN
      DO 420 JB=1,K
  420 Y(JA,JB)=Y(JA,JB)*DY(JB)+YMIN(JB)
      RETURN
      END

      SUBROUTINE EXISA(X,Y,BCD,NC,SIZE,THETA,YMIN,DY)
      DIMENSION BCD(1)
      DATA EPS/1.E-6/
      ZING=1.
      IF(NC) 1,2,2
    1 ZING = -1.
    2 NAC=IABS(NC)
      TH=THETA*.0174533
      N=SIZE+.5
      CTH=COS(TH)
      STH=SIN(TH)
      TN=N
      XB=X
      YB=Y
```

```
C       DRAW AXIS WITH TIC MARKS
        XA=X-.06*ZING*STH
        YA=Y+.06*ZING*CTH
        CALL PLOT(XA,YA,3)
        DO 20 I=1,N
        CALL PLOT(XB,YB,2)
        XC=XB+CTH
        YC=YB+STH
        CALL PLOT(XC,YC,2)
        XA=XA+CTH
        YA=YA+STH
        CALL PLOT(XA,YA,2)
        XB=XC
   20   YB=YC
C         NUMBER THE TIC MARKS
        IF(DY) 22,31,22
   22   CHAR=ABS(YMIN)
        ABSV=ABS(YMIN+DY)
        IF(ABSV-CHAR) 5,6,6
    5   ABSV=CHAR
    6   POW=0.
        CALL ANN(ABSV,BBSV,KPOW)
        POW=KPOW
   93   ADY=DY*10.**(-POW)
        ABSV=YMIN*10.**(-POW)+TN*ADY
        XA=XB-(.18*ZING-.07)*STH-.1714*CTH
        YA=YB+(.18*ZING-.07)*CTH-.1714*STH
        N=N+1
        DO 30 I=1,N
        IF(ABS(ABSV)-EPS) 101,101,100
  100   CALL NUMBER(XA,YA,.14,ABSV,THETA,2)
  101   ABSV=ABSV-ADY
        XA=XA-CTH
   30   YA=YA-STH
   31   TNC=NAC+7
C      WRITE AXIS LABEL
        XA=X+(SIZE*.5-.06*TNC)*CTH-(-.14+ZING*.46)*STH
        YA=Y+(SIZE*.5-.06*TNC)*STH+(-.14+ZING*.46)*CTH
        CALL SYMBOL(XA,YA,.14,BCD,THETA,NAC)
        XA=XA+((TNC-6.)*.12)*CTH
        YA=YA+((TNC-6.)*.12)*STH
  802   IF(DY) 801,50,801
  801   IF(POW) 35,50,35
   35   CALL SYMBOL(XA,YA,.14,7H(X10   ),THETA,7)
        XA=XA+.48*CTH-.14*STH
        YA=YA+.48*STH+.14*CTH
        IF(POW) 40,50,40
   40   CALL NUMBER(XA,YA,.14,POW,THETA,-1)
```

```
   50 RETURN
      END

      SUBROUTINE LIND(X,Y,N,K,L)
      DIMENSION X(1),Y(1)
      DATA IPLUS,ISQ,IDM,ISTAR,ICROSS,IDOT
     C  /'+', ZCA000000, ZCB000000, '*', ZCC000000, '.'/
      I3=3
      KK=K
C...DATA IN EVERY KTH CELL
      IF(K) 5,55,6
   55 WRITE(6,1000)
 1000 FORMAT(5X,'PLT CODE 01  K=0 IN ARG LIST FOR SUBROUT. LINE')
      RETURN
    5 I3=2
      KK=-KK
    6 NP=N*KK
C...TEST BRANCH VALUE
      LL=IABS(L)
      IF(LL)65,65,64
   64 IF(LL-7) 7,7,65
   65 LL=1
    7 DO 10 I=1,NP,KK
      CALL PLOT(X(I),Y(I),I3)
C...PEN CONDITION BETWEEN POINTS
      IF(L)71,71,73
   71 I3=3
      GO TO 199
   73 I3=2
C...BRANCH TO SPECIFIED MARKER
  199 GO TO(10,200,300,400,500,600,700),LL
  200 IBCD=IPLUS
      GO TO 9
  300 IBCD=ISQ
      GO TO 91
  400 IBCD=IDM
      GO TO 9
  500 IBCD=ISTAR
      GO TO 9
  600 IBCD=ICROSS
      GO TO 91
  700 IBCD=IDOT
      XP=X(I)-.03
      YP=Y(I)-.03
      GO TO 94
    9 XP=X(I)-.04
      YP=Y(I)-.06
```

```
      GO TO 94
   91 XP=X(I)-.03
      YP=Y(I)-.05
      GO TO 94
   94 CALL SYMBOL(XP,YP,.14,IBCD,0.,1)
   95 CALL PLOT(X(I),Y(I),3)
   10 CONTINUE
   11 RETURN
      END

      SUBROUTINE CENT(BCD,NTT)
      DIMENSION BCD(1)
      DATA BLNK/'    '/
      NTT=0
      DO 10 JA=1,20
      JB=JA
      IF(BCD(JA).NE.BLNK) GO TO 20
   10 CONTINUE
      RETURN
   20 JA=20
   21 IF(BCD(JA).NE.BLNK) GO TO 30
      JA=JA-1
      IF(JA.LT.0) RETURN
      GO TO 21
   30 JC=JA
   40 NTTA=JC-JB+1
      NTT=4*NTTA
      DO 50 JA=1,NTTA
      JD=JA+JB-1
   50 BCD(JA)=BCD(JD)
      NTTP=NTTA+1
      IF(NTTP.GT.20) RETURN
      DO 60 JA=NTTP,20
   60 BCD(JA)=BLNK
      RETURN
      END

      SUBROUTINE DRUA(X,Y,K,N,XSIZE,YSIZE,NPT)
      DIMENSION X(1),Y(N,1),YMIN(20),DY(20)
      DIMENSION TITLE(20),XNAME(20),YNAME(20,20)
      PAPER=11.
   50 FORMAT(20A4)
      READ(5,50,END=400) (TITLE(JA),JA=1,20)
      READ(5,50,END=400) (XNAME(JA),JA=1,20)
      DO 60 JA=1,K
      READ(5,50,END=400) (YNAME(JB,JA),JB=1,20)
```

```
   60 CONTINUE
      CALL PLOTS
      CALL PLOT(-11.,0.,-3)
      CALL SYMBOL(0.,0.,.21,'BOX 81   WINFREY',0.,15)
      CALL PLOT(0.,4.,-3)
      NSIGN=1
      IF(NPT.LT.0) NSIGN=-1
      NPA=NPT*NSIGN
      XST=FLOAT(K)*.5
      XSTP=XST+(PAPER-XSIZE)*.5
      IF(XSIZE.GT.8.01) XSTP=XST+1.
      XAXIS=XSIZE-XST
      CALL SCALE(X,N,1,XAXIS,0.,XMIN,DX)
      DO 100 JA=1,K
      CALL SCALE(Y(1,JA),N,1,YSIZE,0.,YMIN(JA),DY(JA))
  100 CONTINUE
      DO 130 JA=1,N
  130 X(JA)=X(JA)+XSTP
      YMAX=YMIN(1)+DY(1)*YSIZE
      YS=ABS(YMIN(1))/DY(1)
      IF(YMAX.LT.0.) YS=0.
      IF(YMIN(1). GT.0.) YS=0.
      IF(XSIZE.GT.8.01) GO TO 200
      CALL CENT(XNAME,NTT)
      NLT=-NTT
      CALL EXIS(XSTP,YS,XNAME,NLT,XAXIS,0.,XMIN,DX)
      JB=1
      DO 160 JA=1,K
      JB=JB+1
      IF(JB.EQ.8) JB=2
      YST=XSTP-.5*FLOAT(JA-1)
      CALL CENT(YNAME(1,JA),NTT)
      CALL EXIS(YST,0.,YNAME(1,JA),NTT,YSIZE,90.,YMIN(JA),DY(JA))
      JC=NPT
      IF(NPA.EQ.10) JC=JB*NSIGN
      CALL LINE(X,Y(1,JA),N,1,JC)
  160 CONTINUE
      YL=-.8
      CALL CENT(TITLE,NTT)
      XL=PAPER*.5-FLOAT(NTT/2)*.12
      CALL SYMBOL(XL,  YL,.14,TITLE,0.,NTT)
      YAX=YSIZE+3.
      GO TO 300
  200 XFAR=(PAPER+YSIZE)*.5
      YS=XFAR-YS
      CALL CENT(XNAME,NTT)
      NLT=-NTT
      CALL EXIS(YS,XSTP,XNAME,NLT,XAXIS,90.,XMIN,DX)
```

```
      JB=1
      DO 260  JA=1,K
      JB=JB+1
      IF(JB.EQ.8)  JB=2
      YST=XSTP-.5*FLOAT(JA-1)
      CALL  CENT(YNAME(1,JA),NTT)
      CALL  EXIS(XFAR,YST,YNAME(1,JA),NTT,YSIZE,180.,YMIN(JA),DY(JA))
      DO 230  JC=1,N
  230 Y(JC,JA)=XFAR-Y(JC,JA)
      JC=NPT
      IF(NPA.EQ.10)  JC=JB*NSIGN
      CALL  LINE(Y(1,JA),X,N,1,JC)
      DO 260  JC=1,N
      Y(JC,JA)=XFAR-Y(JC,JA)
  260 CONTINUE
      XL=XFAR+.8
      CALL  CENT(TITLE,NTT)
      YL=XSIZE*.5+1.-FLOAT(NTT/2)*.12
      CALL  SYMBOL(XL,YL,.14,TITLE,90.,NTT)
      YAX=XSIZE+3.
  300 CALL  PLOT(0.,YAX,-3)
      CALL  SYMBOL(0.,0.,.21,'BOX 81  WINFREY',0.,15)
      CALL  PLOT(1.25,4.,-3)
      CALL  PLOTE
      WRITE(6,380)(TITLE(J),J=1,20)
  380 FORMAT(/////20X,75HA PLOT HAS BEEN DRAWN BY SUBROUTINE DRUA CONTAI
     1NING THE FOLLOWING TITLE -        ,////20X,20A4,////)
C    RESTORE ORIGINAL VALUES TO X AND Y
      DO 390  JA=1,N
      X(JA)=(X(JA)-XSTP)*DX+XMIN
      DO 390  JB=1,K
  390 Y(JA,JB)=Y(JA,JB)*DY(JB)+YMIN(JB)
      RETURN
  400 JA=2+K
      WRITE(6,450)  JA
  450 FORMAT(1H1////20X,43HNOT ENOUGH DATA CARDS FOR SUBROUTINE DRUA.
     1  ////I22,52H CARDS ARE READ EACH TIME SUBROUTINE DRUA IS CALLED.
     2     ////20X54HRETURNED TO CALLING ROUTINE WITHOUT STARTING A PLOT.
     3     /1H1//)
      RETURN
      END

      SUBROUTINE DRUB(X,Y,K,N,XSIZE,YSIZE,NPT,LABEL,TITLE,XNAME,YNAME)
      DIMENSION X(1),Y(N,1),YMIN(20),DY(20)
      DIMENSION TITLE(1),XNAME(1),YNAME(20,1),LABEL(1)
      PAPER=11.
      CALL  PLOTS
```

```
      CALL PLOT(-11.,0.,-3)
      CALL SYMBOL(0.,0.,.14,LABEL,0.,80)
      CALL PLOT(0.,4.,-3)
      NSIGN=1
      IF(NPT.LT.0) NSIGN=-1
      NPA=NPT*NSIGN
      XST=FLOAT(K)*.5
      XSTP=XST+(PAPER-XSIZE)*.5
      IF(XSIZE.GT.8.01) XSTP=XST+1.
      XAXIS=XSIZE-XST
      CALL SCALE(X,N,1,XAXIS,0.,XMIN,DX)
      DO 100 JA=1,K
      CALL SCALE(Y(1,JA),N,1,YSIZE,0.,YMIN(JA),DY(JA))
  100 CONTINUE
      DO 130 JA=1,N
  130 X(JA)=X(JA)+XSTP
      YMAX=YMIN(1)+DY(1)*YSIZE
      YS=ABS(YMIN(1))/DY(1)
      IF(YMAX.LT.0.) YS=0.
      IF(YMIN(1). GT.0.) YS=0.
      IF(XSIZE.GT.8.01) GO TO 200
      CALL CENT(XNAME,NTT)
      NLT=-NTT
      CALL EXIS(XSTP,YS,XNAME,NLT,XAXIS,0.,XMIN,DX)
      JB=1
      DO 160 JA=1,K
      JB=JB+1
      IF(JB.EQ.8) JB=2
      YST=XSTP-.50*FLOAT(JA-1)
      CALL CENT(YNAME(1,JA),NTT)
      CALL EXIS(YST,0.,YNAME(1,JA),NTT,YSIZE,90.,YMIN(JA),DY(JA))
      JC=NPT
      IF(NPA.EQ.10) JC=JB*NSIGN
      CALL LINE(X,Y(1,JA),N,1,JC)
  160 CONTINUE
      YL=-.8
      CALL CENT(TITLE,NTT)
      XL=PAPER*.5-FLOAT(NTT/2)*.12
      CALL SYMBOL(XL,  YL,.14,TITLE,0.,NTT)
      YAX=YSIZE+3.
      GO TO 300
  200 XFAR=(PAPER+YSIZE)*.5
      YS=XFAR-YS
      CALL CENT(XNAME,NTT)
      NLT=-NTT
      CALL EXIS(YS,XSTP,XNAME,NLT,XAXIS,90.,XMIN,DX)
      JB=1
      DO 260 JA=1,K
```

```
      JB=JB+1
      IF(JB.EQ.8) JB=2
      YST=XSTP-.50*FLOAT(JA-1)
      CALL CENT(YNAME(1,JA),NTT)
      CALL EXIS(XFAR,YST,YNAME(1,JA),NTT,YSIZE,180.,YMIN(JA),DY(JA))
      DO 230 JC=1,N
  230 Y(JC,JA)=XFAR-Y(JC,JA)
      JC=NPT
      IF(NPA.EQ.10) JC=JB*NSIGN
      CALL LINE(Y(1,JA),X,N,1,JC)
      DO 260 JC=1,N
      Y(JC,JA)=XFAR-Y(JC,JA)
  260 CONTINUE
      XL=XFAR+.8
      IF(YS.LT..01) XL=XL+.55
      CALL CENT(TITLE,NTT)
      YL=XSIZE*.5+1.-FLOAT(NTT/2)*.12
      CALL SYMBOL(XL,YL,.14,TITLE,90.,NTT)
      YAX=XSIZE+3.
  300 CALL PLOT(0.,YAX,-3)
      CALL SYMBOL(0.,0.,.14,LABEL,0.,80)
      CALL PLOT(1.25,4.,-3)
      CALL PLOTE
      WRITE(6,400)(TITLE(J),J=1,20)
  400 FORMAT(/////20X,75HA PLOT HAS BEEN DRAWN BY SUBROUTINE DRUB CONTAI
     1NING THE FOLLOWING TITLE -          ,////20X,20A4,////)
C     RESTORE ORIGINAL VALUES OF X AND Y
      DO 420 JA=1,N
      X(JA)=(X(JA)-XSTP)*DX+XMIN
      DO 420 JB=1,K
  420 Y(JA,JB)=Y(JA,JB)*DY(JB)+YMIN(JB)
      RETURN
      END


      SUBROUTINE DRUC(X,Y,K,N,XSIZE,YSIZE,NPT,LABEL,TITLE,XNAME,YNAME)
      DIMENSION TITLE(1),XNAME(1),YNAME(1),LABEL(1),X(N,1),Y(N,1)
      PAPER=11.
      CALL PLOTS
      CALL PLOT(-11.,0.,-3)
      CALL SYMBOL(0.,0.,.14,LABEL,0.,80)
      CALL PLOT(0.,4.,-3)
      NSIGN=1
      IF(NPT.LT.0) NSIGN=-1
      NPA=NPT*NSIGN
      XST=1.
      XSTP=XST+(PAPER-XSIZE)*.5
      IF(XSIZE.GT.8.01) XSTP=XST+1.
```

```
      XAXIS=XSIZE-XST
      NK=N*K
      CALL SCALE(X,NK,1,XAXIS,0.,XMIN,DX)
      CALL SCALE(Y,NK,1,YSIZE,0.,YMIN,DY)
      DO 130 JB=1,K
      DO 130 JA=1,N
  130 X(JA,JB)=X(JA,JB)+XSTP
      YMAX=YMIN+DY*YSIZE
      YS=ABS(YMIN/DY)
      IF(YMAX.LT.0.) YS=0.
      IF(YMIN.GT.0.) YS=0.
      IF(XSIZE.GT.8.01) GO TO 200
      CALL CENT(XNAME,NTT)
      NLT=-NTT
      CALL EXIS(XSTP,YS,XNAME,NLT,XAXIS,0.,XMIN,DX)
      CALL CENT(YNAME,NTT)
      CALL EXIS(XSTP,0.,YNAME,NTT,YSIZE,90.,YMIN,DY)
      JB=1
      DO 160 JA=1,K
      JB=JB+1
      IF(JB.EQ.8) JB=2
      JC=NPT
      IF(NPA.EQ.10) JC=JB*NSIGN
      CALL LINE(X(1,JA),Y(1,JA),N,1,JC)
  160 CONTINUE
      YL=-.8
      CALL CENT(TITLE,NTT)
      XL=PAPER*.5-FLOAT(NTT/2)*.12
      CALL SYMBOL(XL,  YL,.14,TITLE,0.,NTT)
      YAX=YSIZE+3.
      GO TO 300
  200 XFAR=(PAPER+YSIZE)*.5
      YS=XFAR-YS
      CALL CENT(XNAME,NTT)
      NLT=-NTT
      CALL EXIS(YS,XSTP,XNAME,NLT,XAXIS,90.,XMIN,DX)
      CALL CENT(YNAME,NTT)
      CALL EXIS(XFAR,XSTP,YNAME,     NTT,YSIZE,180.,YMIN,DY)
      JB=1
      DO 260 JA=1,K
      JB=JB+1
      IF(JB.EQ.8) JB=2
      DO 230 JC=1,N
  230 Y(JC,JA)=XFAR-Y(JC,JA)
      JC=NPT
      IF(NPA.EQ.10) JC=JB*NSIGN
      CALL LINE(Y(1,JA),X(1,JA),N,1,JC)
      DO 260 JC=1,N
```

```
      Y(JC,JA)=XFAR-Y(JC,JA)
  260 CONTINUE
      XL=XFAR+.8
      CALL CENT(TITLE,NTT)
      YL=XSIZE*.5+1.-FLOAT(NTT/2)*.12
      CALL SYMBOL(XL,YL,.14,TITLE,90.,NTT)
      YAX=XSIZE+3.
  300 CALL PLOT(0.,YAX,-3)
      CALL SYMBOL(0.,0.,.14,LABEL,0.,80)
      CALL PLOT(1.25,4.,-3)
      CALL PLOTE
      WRITE(6,400)(TITLE(J),J=1,20)
  400 FORMAT(/////20X,75HA PLOT HAS BEEN DRAWN BY SUBROUTINE DRUC CONTAI
     1NING THE FOLLOWING TITLE -       ,////20X,20A4,/////)
C    RESTORE ORIGINAL VALUES TO X AND Y
      DO 420 JA=1,N
      DO 420 JB=1,K
      X(JA,JB)=(X(JA,JB)-XSTP)*DX+XMIN
  420 Y(JA,JB)=Y(JA,JB)*DY+YMIN
      RETURN
      END


      SUBROUTINE EXIS (X,Y,BCD,NC,SIZE,THETA,YMIN,DY)
      DIMENSION BCD(1)
      DATA EPS/1.E-6/
      ZING=1.
      IF(NC) 1,2,2
    1 ZING = -1.
    2 NAC=IABS(NC)
      TH=THETA*.0174533
      N=SIZE+.5
      CTH=COS(TH)
      STH=SIN(TH)
      TN=N
      XB=X
      YB=Y
      XA=X-.1*ZING*STH
      YA=Y+.1*ZING*CTH
      CALL PLOT(XA,YA,3)
      DO 20 I=1,N
      CALL PLOT(XB,YB,2)
      XC=XB+CTH
      YC=YB+STH
      CALL PLOT(XC,YC,2)
      XA=XA+CTH
      YA=YA+STH
      CALL PLOT(XA,YA,2)
```

```
      XB=XC
   20 YB=YC
      IF(DY) 22,31,22
   22 CHAR=ABS(YMIN)
      ABSV=ABS(YMIN+DY)
      IF(ABSV-CHAR) 5,6,6
    5 ABSV=CHAR
    6 POW=0.
      CALL ANN(ABSV,BBSV,KPOW)
      POW=KPOW
   93 ADY=DY*10.**(-POW)
      ABSV=YMIN*10.**(-POW)+TN*ADY
      XA=XB-(.2*ZING-.05)*STH-.0857*CTH
      YA=YB+(.2*ZING-.05)*CTH-.0857*STH
      N=N+1
      DO 30 I=1,N
      IF(ABS(ABSV)-EPS) 101,101,100
  100 CALL NUMBER(XA,YA,.1,ABSV,THETA,2)
  101 ABSV=ABSV-ADY
      XA=XA-CTH
   30 YA=YA-STH
   31 TNC=NAC+7
      XA=X+(SIZE*.5-.03*TNC)*CTH-(-.07+ZING*.36)*STH
      YA=Y+(SIZE*.5-.03*TNC)*STH+(-.07+ZING*.36)*CTH
      CALL SYMBOL(XA,YA,.07,BCD,THETA,NAC)
      XA=XA+((TNC-6.)*.06)*CTH
      YA=YA+((TNC-6.)*.06)*STH
  802 IF(DY) 801,50,801
  801 IF(POW) 35,50,35
   35 CALL SYMBOL(XA,YA,.07,7H(X10   ),THETA,7)
      XA=XA+.24*CTH-.07*STH
      YA=YA+.24*STH+.07*CTH
      IF(POW) 40,50,40
   40 CALL NUMBER(XA,YA,.1,POW,THETA,-1)
   50 RETURN
      END

      SUBROUTINE DRUQ(X,Y,K,N,XSIZE,YSIZE,NPT,LABEL,TITLE,XNAME,YNAME)
      DIMENSION TITLE(1),XNAME(1),YNAME(1),LABEL(1),X(N,1),Y(N,1)
      CALL       DRUCL(X,Y,K,N,XSIZE,YSIZE,NPT,LABEL,TITLE,XNAME,YNAME)
      RETURN
      END
```

```
      SUBROUTINE DRUCL(X,Y,K,N,XSIZE,YSIZE,NPT,LABEL,TITLE,XNAME,YNAME)
      DIMENSION TITLE(1),XNAME(1),YNAME(1),LABEL(1),X(N,1),Y(N,1)
      PAPER=11.
      CALL PLOTS
      CALL PLOT(-11.,0.,-3)
      CALL SYMBOL(0.,0.,.14,LABEL,0.,80)
      CALL PLOT(0.,4.,-3)
      NSIGN=1
      IF(NPT.LT.0) NSIGN=-1
      NPA=NPT*NSIGN
      XST=1.
      XSTP=XST+(PAPER-XSIZE)*.5
      IF(XSIZE.GT.8.01) XSTP=XST+1.
      XAXIS=XSIZE-XST
      NK=N*K
      CALL SCALE(X,NK,1,XAXIS,0.,XMIN,DX)
      CALL SCALE(Y,NK,1,YSIZE,0.,YMIN,DY)
      DO 130 JB=1,K
      DO 130 JA=1,N
  130 X(JA,JB)=X(JA,JB)+XSTP
      YMAX=YMIN+DY*YSIZE
      YS=ABS(YMIN/DY)
      IF(YMAX.LT.0.) YS=0.
      IF(YMIN.GT.0.) YS=0.
      IF(XSIZE.GT.8.01) GO TO 200
      CALL CENT(XNAME,NTT)
      NLT=-NTT
      CALL EXISA(XSTP,YS,XNAME,NLT,XAXIS,0.,XMIN,DX)
      CALL CENT(YNAME,NTT)
      CALL EXISA(XSTP,0.,YNAME,NTT,YSIZE,90.,YMIN,DY)
      JB=1
      DO 160 JA=1,K
      JB=JB+1
      IF(JB.EQ.8) JB=2
      JC=NPT
      IF(NPA.EQ.10) JC=JB*NSIGN
      CALL LIND(X(1,JA),Y(1,JA),N,1,JC)
      CALL LIND(X(1,JA),Y(1,JA),N,1,JC)
      CALL LIND(X(1,JA),Y(1,JA),N,1,JC)
  160 CONTINUE
      YL=-.8
      CALL CENT(TITLE,NTT)
      XL=PAPER*.5-FLOAT(NTT/2)*.18
      CALL SYMBOL(XL,  YL,.21,TITLE,0.,NTT)
      YAX=YSIZE+3.
      GO TO 300
  200 XFAR=(PAPER+YSIZE)*.5
      YS=XFAR-YS
```

```
      CALL CENT(XNAME,NTT)
      NLT=-NTT
      CALL EXISA(YS,XSTP,XNAME,NLT,XAXIS,90.,XMIN,DX)
      CALL CENT(YNAME,NTT)
      CALL EXISA(XFAR,XSTP,YNAME,      NTT,YSIZE,180.,YMIN,DY)
      JB=1
      DO 260 JA=1,K
      JB=JB+1
      IF(JB.EQ.8) JB=2
      DO 230 JC=1,N
  230 Y(JC,JA)=XFAR-Y(JC,JA)
      JC=NPT
      IF(NPA.EQ.10) JC=JB*NSIGN
      CALL LIND(Y(1,JA),X(1,JA),N,1,JC)
      CALL LIND(Y(1,JA),X(1,JA),N,1,JC)
      CALL LIND(Y(1,JA),X(1,JA),N,1,JC)
      DO 260 JC=1,N
      Y(JC,JA)=XFAR-Y(JC,JA)
  260 CONTINUE
      XL=XFAR+.8
      CALL CENT(TITLE,NTT)
      YL=XSIZE*.5+1.-FLOAT(NTT/2)*.18
      CALL SYMBOL(XL,YL,.21,TITLE,90.,NTT)
      YAX=XSIZE+3.
  300 CALL PLOT(0.,YAX,-3)
      CALL SYMBOL(0.,0.,.14,LABEL,0.,80)
      CALL PLOT(1.25,4.,-3)
      CALL PLOTE
      WRITE(6,400)(TITLE(J),J=1,20)
  400 FORMAT(/////20X,75HA PLOT HAS BEEN DRAWN BY SUBROUTINE DRUQ CONTAI
     1NING THE FOLLOWING TITLE -       ,////20X,20A4,////)
C   RESTORE ORIGINAL VALUES TO X AND Y
      DO 420 JA=1,N
      DO 420 JB=1,K
      X(JA,JB)=(X(JA,JB)-XSTP)*DX+XMIN
  420 Y(JA,JB)=Y(JA,JB)*DY+YMIN
      RETURN
      END
```

```
      SUBROUTINE DTPRIM( RECENT, KK, OLD, DT, DTP, NPRINT)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER YES
C
      DIMENSION DTP(1)
C
      COMMON NUNIT, IPRINT
C
      DATA YES/4HYES /, NO/4HNO  /
C
 5001 FORMAT(' ', 10X, 'FOR INTERATION ', I1, ' RECENT = ',
     1       1PD15.7, ' AND OLD = ', D15.7, '   OLD - RECENT =', D15.7)
C
      SMALL = 1.0D-10
C
C
      DIFF = OLD - RECENT
      IF(DIFF .LT. SMALL)  DIFF = SMALL
C
      DTP(KK) = ( OLD/ DIFF) * DT
C
      IF(NPRINT .EQ. NO)  RETURN
C
      WRITE(NUNIT,5001) KK, RECENT, OLD, DIFF
C
      RETURN
      END
```

```
C
C-----------------------------------------------------------------------
C
C                SUBROUTINE CODED BY DR. R. C. WINFREY
C
C-----------------------------------------------------------------------
C

      SUBROUTINE DVVS(A,B,ELAM,X,N,NA,NB,NX,KSYS,NERR)
C
C     THIS ROUTINE FINDS THE EIGENVALUES AND ON OPTION THE
C          EIGENVECTORS OF THE SYSTEM (A-ELAM*B)X=0
C
C     A=        SYMMETRIC MATRIX--DESTROYED BY ROUTINE
C     B=        POSITIVE DEFINITE MATRIX--DESTROYED BY ROUTINE
C     ELAM=     VECTOR WHERE EIGENVALUES WILL BE STORED
C     X=        MATRIX WHERE EIGENVECTORS WILL BE STORED BY COLUMNS
C     N=        ORDER OF MATRICES A, B AND X AND VECTOR ELAM
C     NA=       ROW DIMENSION OF MATRIX A(GT OR =N)
C     NB=       ROW DIMENSION OF MATRIX B(GT OR =N)--IF B IS IDENTITY
C                SET NB=0, THEN STORAGE FOR B IS NOT NECESSARY
C     NX=       ROW DIMENSION OF MATRIX X(GT OR =N)--IF VECTORS ARE NOT
C                DESIRED SET NX=0, THEN STORAGE FOR X IS NOT NECESSARY
C     NERR=    RETURN INDICATOR
C                =0, NORMAL RETURN
C                =NEGATIVE, SOLUTION DID NOT CONVERGE
C                =POSITVE, B IS NOT POSITIVE DEFINITE AND NERR CONTAINS
C                 THE INDEX OF DIAGONAL ELEMENT CAUSING THIS CONDITION
C
      DOUBLE PRECISION A(KSYS,1),B(KSYS,1),X(KSYS,KSYS),ELAM(KSYS)
      DOUBLE PRECISION TBI,T1,T2,T3,TL1,TAI,TBJ,TLJ,S,C,TLI,EMAX
      TL1=0.
      NN=N
      NERR=0
      NNX=NX
C
C     IF EIGENVECTORS ARE REQUESTED STORE IDENTITY AT X
C
      IF (NNX.EQ.0)GO TO 20
      DO 15 I=1,NN
      DO 10 J=1,NN
 10   X(I,J)=0.0
 15   X(I,I)=1.0
C
C     IF B=I MATRIX REDUCTION NOT NECESSARY
C
 20   IF (NB.EQ.0)GO TO 200
```

```
C
C     MATRIX REDUCTION---FIND MATRIX M SUCH THAT M(T)BM=I
C          THE ACTUAL COMPUTATION OF M(UPPER TRIANGULAR) IS ONLY
C          DONE WHEN EIGENVECTORS ARE REQUESTED IT IS STORED AT X
C     REDUCE B TO I(UPPER TRIANGULAR)
C     REDUCE A TO A1=M(T)AM(LOWER TRIANGULAR)
C     THE EIGENVALUES OF (A-ELAM*B)X=0 EQUAL THOSE OF (A1-ELAM*I)Y=0
C     THE EIGENVECTORS HAVE THE RELATION X=YM
C
      DO 150 I=1,NN
      TBI=B(I,I)
      IF(TBI.GT.0.0) GO TO 100
      NERR=I
      RETURN
  100 T1=1.0/TBI
      TLI=DSQRT(T1)
      TAI=A(I,I)
      IM1=I-1
      IF(I.EQ.NN) GO TO 130
      IP1=I+1
      DO 125 J=IP1,NN
      TBJ=B(I,J)
      ELAM(J)=-TBJ*T1
      TLJ=ELAM(J)
      IF(IM1.EQ.0) GO TO 110
      DO 105 K=1,IM1
  105 A(J,K)=TLJ*A(I,K)+A(J,K)
  110 T2=TLJ*TAI+A(J,I)
      DO 115 K=IP1,J
      B(K,J)=B(K,J)+ELAM(K)*TBJ
  115 A(J,K)=ELAM(K)*T2+TLJ*A(K,I)+A(J,K)
      IF (NNX.EQ.0)GO TO 125
C
C     IF NO VECTORS ARE REQUIRED M IS NOT COMPUTED
C
      DO 120 K=1,I
  120 X(K,J)=X(K,I)*TLJ+X(K,J)
  125 CONTINUE
      DO 127 J=IP1,NN
  127 A(J,I)=TLI*(ELAM(J)*TAI+A(J,I))
      IF(IM1.EQ.0) GO TO 140
  130 DO 135 K=1,IM1
  135 A(I,K)=TLI*A(I,K)
  140 IF (NNX.EQ.0)GO TO 150
      DO 145 K=1,I
  145 X(K,I)=X(K,I)*TLI
  150 A(I,I)=T1*TAI
C
```

```
C
C     SEQUENTIAL THRESHOLD JACOBI METHOD
C
 200  NM1=NN-1
C
C     INITIALIZE THRESHOLD--FIND MAXIMUM OFF DIAGONAL ELEMENT
C
      IS=6
      EMAX=0.0
      DO 215 J=1,NM1
      JP1=J+1
      DO 215 I=JP1,NN
      T1=DABS(A(I,J))
      IF(T1.GT.EMAX) EMAX=T1
 215  CONTINUE
      EMAX=EMAX/4.0
C
C     DIAGONALIZATION SWEEP
C
 300  IST=0
      DO 350 J=1,NM1
      JP1=J+1
      DO 350 I=JP1,NN
      T1=A(I,J)
      IF(DABS(T1).LE.EMAX)GO TO 350
      IST=IST+1
C
C     FORM ROTATION MATRIX ELEMENTS C AND S
C
      T2=A(J,J)-A(I,I)
      IF(T2.EQ.0.0)GO TO 310
      T2=2.0*(T1/T2)
      IF(DABS(T2).GE.1.D9)GO TO 310
      T2=T2/(1.0+DSQRT(1.D0+T2*T2))
      C=1.0/DSQRT(1.D0+T2*T2)
      S=C*T2
      GO TO 320
  310 C=1./DSQRT(2.D0)
      IF(T1.LE.0.0)GO TO 315
      S=C
      GO TO 320
 315  S=-C
C
C     REDUCE A BY ROTATION USING COLUMNS AND ROWS I AND J
C
 320  IM1=I-1
      T3=C*A(I,J)-S*A(J,J)
      A(J,J)=C*A(J,J)+S*A(I,J)
      IF(I.EQ.JP1) GO TO 327
```

```
      DO 325 K=JP1,IM1
      T1=A(K,J)
      T2=A(I,K)
      A(K,J)=C*T1+S*T2
  325 A(I,K)=C*T2-S*T1
  327 DO 330 K=I,NN
      T1=A(K,J)
      T2=A(K,I)
      A(K,J)=C*T1+S*T2
  330 A(K,I)=-S*T1+C*T2
      DO 335 K=1,J
      T1=A(J,K)
      T2=A(I,K)
      A(J,K)=C*T1+S*T2
  335 A(I,K)=-S*T1+C*T2
      A(I,I)=-S*T3+C*A(I,I)
C
C     FORM EIGENVECTORS IF REQUESTED
C
      IF (NNX.EQ.0)GO TO 350
      DO 345 K=1,NN
      T1=X(K,J)
      T2=X(K,I)
      X(K,J)=C*T1+S*T2
  345 X(K,I)=-S*T1+C*T2
  350 CONTINUE
C
C     TEST FOR CONVERGENCE AND MODIFY THRESHOLD
C
      IS=IS-1
      IF(IS.LE.0)GO TO 400
      K=-(2**(5-IS))
      EMAX=EMAX*(2.0**K)
      GO TO 300
  400 IF(IST.EQ.0)GO TO 500
      IF(IS.GE.(-33))GO TO 300
C
C     METHOD HAS FAILED TO CONVERGE
C
      NERR=-1
C
C     TRANSFER EIGENVALUES AND EXIT
C
  500 DO 510 I=1,NN
  510 ELAM(I)=A(I,I)
      RETURN
      END
```

```
      SUBROUTINE FIXFOR(C, F)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER C, YES
C
      DIMENSION F(1)
C
      COMMON NUNIT, IPRINT
C
      DATA  YES/4HYES /, NO/4HNO  /
C
C
      GO TO (10, 20, 30, 40, 50, 60, 70, 80), C
C
   10 CONTINUE
      F(1) = 0.0D0
      F(2) = 0.0D0
      F(7) = 0.0D0
      F(8) = -F(2)
      RETURN
C
   20 CONTINUE
      F(1) = 0.0D0
      F(2) = 0.0D0
      RETURN
C
   30 CONTINUE
      F(1) = 0.0D0
      F(7) = 0.0D0
      F(8) = -F(2)
      RETURN
C
   40 CONTINUE
      F(1) = 0.0D0
      F(8) = -F(2)
      RETURN
C
   50 CONTINUE
      F(7) = 0.0D0
      F(2) = 0.0D0
      F(8) = 0.0D0
      RETURN
C
   60 CONTINUE
      F(2) = 0.0D0
```

```
      F(8) = 0.0D0
      RETURN
C
   70 CONTINUE
      F(7) = 0.0D0
      F(8) = -F(2)
      RETURN
C
   80 CONTINUE
      F(8) = -F(2)
      RETURN
      END
```

```
      SUBROUTINE ADJUST(I1,I2,QR,QRD,QRDD,Q,QD,QDD)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER YES
C
      DIMENSION QR(1), QRD(1), QRDD(1), Q(1), QD(1), QDD(1)
C
      COMMON NUNIT, IPRINT
C
      DATA  YES/4HYES /, NO/4HNO  /
C
      QR  (I1) = Q  (I2)
      QRD (I1) = QD (I2)
      QRDD(I1) = QDD(I2)
C
      RETURN
      END
```

```
      SUBROUTINE GPRINT(FPLOT, DEG, KTIME, YNAME, XSCALE, IP, NPRINT,
     1                  GRID, LINES, TITLE, LABEL)
C
      INTEGER*2 GRID(LINES,1)
C
      INTEGER YES
C
      DIMENSION YNAME(20)   , FPLOT(KTIME)   , DEG(1), XSCALE(1),
     1          TITLE(1), LABEL(1)
C
      COMMON NUNIT, IPRINT
C
      DATA YES/4HYES /, NO/4HNO  /
C
 5000 FORMAT('0',5X,4(1PE15.7, 5X)  )
 5001 FORMAT(' ',5X,I5,5X,1P2E15.7)
 5026 FORMAT('1', 18A4)
 5027 FORMAT('+',72X, 15A4)
 5028 FORMAT('0', 40X, 18A4)
      XMIN = DEG(1)
      XMAX = DEG(KTIME)
      YMIN =  1.0E40
      YMAX=-1.E40
C
      DO  10  K = 1, KTIME
      IF(FPLOT(K) .GT. YMAX)  YMAX = FPLOT(K)
      IF(FPLOT(K) .LT. YMIN)  YMIN = FPLOT(K)
   10 CONTINUE
      WRITE(NUNIT,5026) (LABEL(JA), JA = 1,18)
      WRITE(NUNIT,5027)  (TITLE(JA), JA = 1, 15)
      WRITE(NUNIT,5028) ( YNAME(JA), JA = 1,18 )
C
      IRYP1 = 80*IP + 1
      CALL OLPLOT(DEG,FPLOT,KTIME,XMAX,XMIN,YMAX,YMIN,IRYP1,GRID,XSCALE,
     1            0, NPRINT, LINES)
C
      IF(NPRINT .EQ. NO)  RETURN
C
      WRITE(6,5000) XMIN,XMAX,YMIN,YMAX
C
      DO  51  JA = 1, KTIME
      WRITE(6,5001) JA, DEG(JA), FPLOT(JA)
   51 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE GRAPH(FPLOT, YNAME, XNAME, DEG1, DEG2, Y1, Y2, Y3, Y4,
     1                 YNAME1, YNAME2, PIN2$8, CLEAR,
     2           NTIME, LABEL, TITLE, IDRAW, IPLOT, XSCALE,
     3           IP, NPRINT, GRID, LINES)
C
      INTEGER*2 GRID(LINES,1)
      INTEGER YES, PIN2$8
C
      REAL*8 CLEAR
      REAL*4 LABEL
C
      DIMENSION XNAME(1), DEG1(1), DEG2(NTIME,2), XSCALE(1), TITLE(1),
     1          FPLOT(NTIME,1), YNAME(20,1), Y1(NTIME,1), Y2(NTIME,1),
     2          Y3(NTIME,1), Y4(NTIME,1),
     3          YNAME1(20,1), YNAME2(20,1), LABEL(1)
C
      CCMMON NUNIT, IPRINT
C
      DATA YES/4HYES /, NO/4HNO  /
C
      DO  5  I = 1,10
      IF(FPLOT(1,I) .EQ. 0.0)  FPLOT(1,I) = 1.0E-10
    5 CONTINUE
C
      IF(IDRAW .EQ. NO)   GO TO 2501
C
C
 4201 CALL  DRUB  (DEG1,Y1,2,NTIME,10.,6.,1,LABEL,TITLE,XNAME,YNAME1)
 4202 CALL  DRUB  (DEG1,Y2,2,NTIME,10.,6.,1,LABEL,TITLE,XNAME,YNAME2)
      CALL  DRUC  (DEG2,Y3,2,NTIME,10.,6.,1,LABEL,TITLE,XNAME,YNAME(1,12))
 4203 CALL  DRUPB(DEG1,FPLOT(1,5),1,NTIME,10.,6.,1,LABEL,TITLE,XNAME,
     1          YNAME(1,5) )
 4204 CALL  DRUPB(DEG1,FPLOT(1,6),1,NTIME,10.,6.,1,LABEL,TITLE,XNAME,
     1          YNAME(1,6) )
 4205 CALL  DRUPB(DEG1,FPLOT(1,7),1,NTIME,10.,6.,1,LABEL,TITLE,XNAME,
     1          YNAME(1,7) )
C
      IF(PIN2$8 .EQ. YES)  GO TO 4207
C
 4206 CALL  DRUPB(DEG1,FPLOT(1,9),1,NTIME,10.,6.,1,LABEL,TITLE,XNAME,
     1          YNAME(1,9) )
C
 4207 IF(CLEAR .LE. -10.0)  GO TO 2501
C
 4208 CALL  DRUPB(DEG1,FPLOT(1,10),1,NTIME,10.,6.,1,LABEL,TITLE,XNAME,
     1          YNAME(1,10) )
```

```
 4209 CALL DRUPB(DEG1,FPLOT(1,11),1,NTIME,10.,6.,1,LABEL,TITLE,XNAME,
     1          YNAME(1,11) )
      CALL DRUC (DEG2,Y4,2,NTIME,10.,6.,1,LABEL,TITLE,XNAME,YNAME(1,13))
C
C
C
 2501 CONTINUE
C
C
C
      IF(IPLOT .EQ. NO) GO TO 4199
C
      IF(PIN2$8 .EQ.YES)  GO TO 4102
 4101 CALL GPRINT(FPLOT(1, 1), DEG1, NTIME, YNAME(1, 1), XSCALE, IP,
     1                NPRINT, GRID, LINES, TITLE, LABEL)
C
 4102 CALL GPRINT(FPLOT(1, 2), DEG1, NTIME, YNAME(1, 2), XSCALE, IP,
     1                NPRINT, GRID, LINES, TITLE, LABEL)
 4103 CALL GPRINT(FPLOT(1, 3), DEG1, NTIME, YNAME(1, 3), XSCALE, IP,
     1                NPRINT, GRID, LINES, TITLE, LABEL)
 4104 CALL GPRINT(FPLOT(1, 4), DEG1, NTIME, YNAME(1, 4), XSCALE, IP,
     1                NPRINT, GRID, LINES, TITLE, LABEL)
 4105 CALL GPRINT(FPLOT(1, 5), DEG1, NTIME, YNAME(1, 5), XSCALE, IP,
     1                NPRINT, GRID, LINES, TITLE, LABEL)
 4106 CALL GPRINT(FPLOT(1, 6), DEG1, NTIME, YNAME(1, 6), XSCALE, IP,
     1                NPRINT, GRID, LINES, TITLE, LABEL)
C
      IF(CLEAR  .LE. -10.0)  GO TO 4200
 4107 CALL GPRINT(FPLOT(1, 7), DEG1, NTIME, YNAME(1, 7), XSCALE, IP,
     1                NPRINT, GRID, LINES, TITLE, LABEL)
C
 4200 IF(PIN2$8 .EQ.YES)  GO TO 4109
 4108 CALL GPRINT(FPLOT(1, 8), DEG1, NTIME, YNAME(1, 8), XSCALE, IP,
     1                NPRINT, GRID, LINES, TITLE, LABEL)
C
 4109 CALL GPRINT(FPLOT(1, 9), DEG1, NTIME, YNAME(1, 9), XSCALE, IP,
     1                NPRINT, GRID, LINES, TITLE, LABEL)
C
      IF(CLEAR  .LE. -10.0)  GO TO 4199
 4110 CALL GPRINT(FPLOT(1,10), DEG1, NTIME, YNAME(1,10), XSCALE, IP,
     1                NPRINT, GRID, LINES, TITLE, LABEL)
C
C
 4199 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE INITAL(KSYS,Q,QD,QDD,AMPW,FS,FZERO,XZERO,ROCKER,ISYS,
     1              COLM,COLC,COLK,DP,F,DAMP,STF,STFRI,NPRINT)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER YES
C
      DIMENSION COLM(KSYS), COLC(KSYS), COLK(KSYS)
      DIMENSION Q(10), QD(1), QDD(1), DP(1), F(1), DAMP(KSYS,KSYS),
     1          STFRI(ISYS,1), STF(KSYS,KSYS)
C
      CCMMON NUNIT, IPRINT
C
      DATA YES/4HYES /, NO/4HNO  /
C
 5043 FORMAT('0', 10X, ' FROM SUBROUTINE INITAL')
 5044 FORMAT(' ',  1P7D18.9,//  )
 5045 FORMAT('0',//,20X,'- - - - -  STFI - - - - - - ',//)
 5046 FORMAT('0',//,20X,'- - - -  STFRI - - - - -  ', //)
C               *********************************************
C               **                                         **
C               **          SET UP INITIAL VALUES          **
C               **    OF DISPLACEMENT(Q), VELOCITY(QDD),   **
C               **          AND ACCELERATION(QDD)          **
C               **                                         **
C               *********************************************
C
      DO 300 JA = 1,KSYS
      QD(JA) = -AMPW
      QDD(JA) = 0.0D0
  300 CONTINUE
C
      QD(1) = +AMPW
      QD(2) = +AMPW
      QD(4) = -AMPW*ROCKER
      QD(7) = +AMPW*ROCKER
      QD(8) = +AMPW
C
      Q(1) = XZERO
C
      DO 350 JA = 1,ISYS
      DP(JA) = -(COLM(JA)*QDD(1)+COLC(JA)*QD(1)+COLK(JA)*Q(1))
C
      DO 350 JB = 1,ISYS
      DP(JA) = DP(JA)-DAMP(JA,JB)*QD(JB+1)
      STFRI(JA,JB) = STF (JA,JB)
```

```
C
  350 CONTINUE
C
      DO  8  I = 1, KSYS
      F(I) = 0.0D0
    8 CONTINUE
C
C              *********************************************
C              **                                         **
C              **          CALL SUBROUTINE DINVS           **
C              **  DINVS CALCULATES THE INVERSE STIFFNESS  **
C              **                MATRIX                   **
C              **       SEND STF  - - RETURN STFRI        **
C              **                                         **
C              *********************************************
C
      CALL DINVS(STFRI,ISYS,ISYS)
C
      DO 360 JA = 1,ISYS
      Q(JA+1) = 0.
C
      DO 360 JB = 1,ISYS
      Q(JA+1) = Q(JA+1)+STFRI(JA,JB)*DP(JB)
C
  360 CONTINUE
C
      CALL ADJUST(1,9, Q, QD, QDD, Q, QD, QDD)
      CALL ADJUST(8,2, Q, QD, QDD, Q, QD, QDD)
C
      RETURN
      END
```

```
      SUBROUTINE INPUT(LABEL, TITLE, XNAME, YNAME,
     1                        AGTOUT, CARDS, TAPE, IPLOT, IDRAW,
     2          KPRINT, DAMPNG, RPM, CLEAR, WTDEG, AMP, DELT,
     3          ROCKER, AXILTH, REALTH,
     4          NTIME, KSYS, NCLEAR, NC,
     5          NCOUNT, IDEG, PIN2$8, MPRINT)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      REAL*4 LABEL,TITLE,XNAME,YNAME,
     1       VLV, TWO1, TWO2
C
      INTEGER YES,AGTOUT,DAMPNG,TAPE,CARDS,SEAT,PIN2$8
C
      DIMENSION YNAME(20,1), AXILTH(1), REALTH(1),
     1          LABEL(1), TITLE(1), XNAME(1)
C
      COMMON NUNIT, IPRINT
      DATA YES/4HYES /, NO/4HNO  /
      DATA  TWO1/'   T'/,  TWO2/'WO C'/,  VLV/'NO V'/
C
 5001 FORMAT(18A4)
 5002 FORMAT(1A4)
 5003 FORMAT(D10.3)
 5004 FORMAT(I4)
 5005 FORMAT(1A4,2I4)
 5006 FORMAT(4D10.3)
 5007 FORMAT('0',10(I4,8X) )
 5008 FORMAT('0',6(D10.3,5X) )
 5009 FORMAT('0', 8(A4,10X ) )
 5010 FORMAT(D10.3, 2X, 1A4)
C
      REALTH(1) =  9.0D0
      REALTH(2) =  1.0D0
      REALTH(3) =  1.5D0
      REALTH(4) = 36.0D0
      AXILTH(1) =  1.0D0
      AXILTH(2) =  3.4D0
      AXILTH(3) =  5.1D0
      AXILTH(4) =  1.0D0
      AMP = 0.5D0
      DAMPNG = YES
      KSYS = 8
      NC = 8
      NCLEAR = 3
      NUNIT = 6
```

```
C
      READ(5,5001,END=50)(LABEL(JA), JA = 1, 18)
      READ(5,5001)       (TITLE (JA), JA = 1, 18)
      READ(5,5003) RPM
      READ(5,5010) CLEAR, PIN2$8
      READ(5,5002) MPRINT
      READ(5,5002) IDRAW
      READ(5,5002) AGTOUT
      READ(5,5001) CARDS, TAPE
      READ(5,5003) WTDEG
      READ(5,5005) KPRINT, NCOUNT
      READ(5,5002) IPRINT
      READ(5,5002) IPLOT
      READ(5,5004) IDEG
      READ(5,5003) DELT
      READ(5,5001)       (YNAME(JA,1 ),JA = 1, 18)
      READ(5,5001)       (YNAME(JA,2 ),JA = 1, 18)
      READ(5,5001)       (YNAME(JA,3 ),JA = 1, 18)
      READ(5,5001)       (YNAME(JA,4 ),JA = 1, 18)
      READ(5,5001)       (YNAME(JA,5 ),JA = 1, 18)
      READ(5,5001)       (YNAME(JA,6 ),JA = 1, 18)
      READ(5,5001)       (YNAME(JA,7 ),JA = 1, 18)
      READ(5,5001)       (YNAME(JA,8 ),JA = 1, 18)
      READ(5,5001)       (YNAME(JA,9 ),JA = 1, 18)
      READ(5,5001)       (YNAME(JA,10),JA = 1, 18)
      READ(5,5001)       (YNAME(JA,11),JA = 1, 18)
      READ(5,5001)       (YNAME(JA,12),JA = 1, 18)
      READ(5,5001)       (YNAME(JA,13),JA = 1, 18)
      READ(5,5001)       (XNAME (JA), JA = 1, 18)
C
C
C
C
      ROCKER = REALTH(3) / REALTH(2)
      IF(CLEAR .GT. -10.0)  GO TO 20
      TITLE(4) = VLV
C
   20 IF(PIN2$8 .EQ. NO)  GO TO 25
      LABEL(10) =  TWO1
      LABEL(11) =  TWO2
C
   25 WRITE(NUNIT,5009) AGTOUT,CARDS,TAPE,IPLOT,IDRAW,DAMPNG,PIN2$8,
     1                  MPRINT
      WRITE(NUNIT,5007) NUNIT,NTIME,KSYS,NCLEAR,NC,NCOUNT,IDEG
      WRITE(NUNIT,5008) RPM,CLEAR,WTDEG,AMP,DELT,ROCKER
      WRITE(NUNIT,5008) (AXILTH(I), I = 1,4)
      WRITE(NUNIT,5008) (REALTH(I), I = 1,4)
C
```

```
      RETURN
C
C
   50 IF( IUNIT .EQ. 4 )  END FILE 4
      IF( IUNIT .EQ. 4 )  END FILE 4
      IF( IUNIT .EQ. 4 )  REWIND 4
C
      STOP
C
C
C
C
C
C
C------------------------------------------------------------------------
      END
```

```
      SUBROUTINE MATRIX( COLM, COLC, COLK, SYSM, DAMP, STF, STFI, KSYS,
     1                   KSYSM1, C, IQ, MPRINT, LAYER, IDIM)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER YES, C
C
      DIMENSION COLM(KSYS,1), COLC(KSYS,1), COLK(KSYS,1),
     1          DAMP(KSYS,KSYS,1), STF(KSYS,KSYS,1), STFI(KSYSM1,1),
     2          SYSM(KSYS,KSYS,1), IQ(KSYS,1), LAYER(1), IDIM(1)
C
      COMMON NUNIT, IPRINT
C
      DATA YES/4HYES /, NO/4HNO  /
C
C
      KCOUNT = 1
      LC = LAYER(C)
C              ****************************************
C              **                                    **
C              **   ESTABLISH THE 1X6 COLUMN MATRIX FROM  **
C              **  THE SYSTEM M-C-K MATRICS WITHOUT ROW 1 **
C              **                                    **
C              ****************************************
C
C
C              ****************************************
C              **                                    **
C              **     ESTABLISH THE 6X6 MATRICS FROM THE  **
C              ** SYSTEM M-C-K MATRICS WITHOUT ROW/COL 7  **
C              **                                    **
C              ****************************************
C
   20 DO 200 JA = 1,KSYS
      JC = IQ(JA,C)
      IF(JC .EQ. 0)  GO TO 300
C
      DO 100 JB = 1,KSYS
      JD = IQ(JB,C)
      IF(JD .EQ. 0)  GO TO 150
C
      SYSM(JA,JB,C) = SYSM(JC,JD,LC)
      DAMP(JA,JB,C) = DAMP(JC,JD,LC)
      STF (JA,JB,C) = STF (JC,JD,LC)
  100 CONTINUE
  150 IDIM(C) = KCOUNT
```

```
      KCOUNT = KCOUNT + 1
C
      JCIQ = JC
      IFOR = 1
C
      IF(C .LE. 4)  IFOR = 7
      IF(C .EQ. 1)  JCIQ = IQ(JA,6)
      IF(C .EQ. 3)  JCIQ = IQ(JA,8)
C
      COLM(JA,C) = SYSM(JCIQ,IFOR,LC)
      COLC(JA,C) = DAMP(JCIQ,IFOR,LC)
      COLK(JA,C) = STF (JCIQ,IFOR,LC)
C
C
  200 CONTINUE
  300 IF(MPRINT .EQ. NO)  GO TO 350
C
      CALL PRNTMX( COLM(1,C), COLC(1,C), COLK(1,C), SYSM(1,1,C),
     1             DAMP(1,1,C), STF(1,1,C), KSYS, C, IDIM(C)  )
C
C
  350 IF(C .GT. 1)  RETURN
C
      DO 400 JA = 1,KSYSM1
C
         DO 400 JB = 1,KSYSM1
         STFI(JA,JB) = STF(JA,JB,3)
C
  400 CONTINUE
C
C            ****************************************
C            **                                    **
C            **         CALL SUBROUTINE DINVS       **
C            **  DINVS CALCULATES THE INVERSE STIFFNESS  **
C            **              MATRIX                **
C            **       SEND STF - - RETURN STFI     **
C            **                                    **
C            ****************************************
C
      CALL DINVS(STFI,KSYSM1,KSYSM1)
C
      RETURN
      END
```

```
C
C-----------------------------------------------------------------------
C
C         SUBROUTINE CODED BY DR. R. C. WINFREY AND R. ANDERSON
C                     MODIFIED BY C. W. GNILKA
C
C-----------------------------------------------------------------------
C

      SUBROUTINE MCK( SYSM, DAMP, STF, KSYS, JC, N, NPRINT, COFD, PHI,
     1                ELAM, C)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER YES, C
C
      DIMENSION PHTINV(8,8),PHITM(8,8), PHITK(8,8), DIAGM(8,8),
     1          E(5), SEE(8,8), PHINV(8,8), PHI(KSYS,KSYS,1),
     2          SYSM(KSYS,KSYS,1), STF(KSYS,KSYS,1), DAMP(KSYS,KSYS,1),
     3          DIAGK(8,8), DIAGC(8,8), ZTA(8), BETA(8,8), L(8), Y(5),
     4          AM(5), AMM(5), DIA(5), ELAM(KSYS,1), PHTIDC(8,8)
C
      COMMON NUNIT, IPRINT
C
      DATA YES/4HYES /, NO/4HNO  /
C
C     LENGTH
      DATA  Y/9.0D0, 1.0D0, 1.5D0, 36.0D0, 3.0D0/
C
C
      DATA  AMM/5*0.283D0/,   E/5*3.0D7/
C
      DATA  DIA/0.3D0, 0.6D0, 0.6D0, 0.07D0, 0.3D0/
C
      DATA  PI/3.14159265359D0/, GRAV/386.0D0/
C
C
  330 FORMAT('0',//,10X,'ERROR RETURN FROM DVVS. NERR = ', I3,/// )
  700 FORMAT(' ',4X, 8(1PD15.8, 1X )  )
C
C
C
      IF(NPRINT .EQ. YES) WRITE(NUNIT,700) (ZTA(KA), KA = 1,JC)
      RAD = PI/180.0D0
C     N IS THE NUMBER OF ELEMENTS
C     JC = SIZE OF K (OR M) AND NUMBER OF COL OF B
C
```

```
C
      DO 5 I = 1, N
      AM(I) = AMM(I) * PI * DIA(I)**2 * Y(I) /(4.0D0 * GRAV)
    5 CONTINUE
C
      DO 10  I = 1, JC
      ZTA(I) = 0.1D0
C
      IF(C .EQ. 1)  ZTA(4) = 0.5D0
      IF(C .EQ. 3)  ZTA(1) = 0.5D0
      IF(C .EQ. 3)  ZTA(4) = 0.1D0
C     SET SYSTEM K (OR M) TO 0
C
      DO 10  J = 1, JC
      SYSM(I,J,C) = 0.0D0
      STF(I,J,C) = 0.0D0
   10 CONTINUE
C
C     CALCULATE  EACH  ELEMENT K (OR M)
      APYX = PI * E(3) * DIA(3)**4 / 64.0D0
      APYY = PI * E(2) * DIA(2)**4 / 64.0D0
C
      ONE   = E(1) * PI * DIA(1)**2 /( Y(1)*4.0D0 )
      TWO   = 12.0D0 * APYY /( Y(2)**3 )
      THREE =  6.0D0 * APYY /( Y(2)**2 )
      FOUR  =  4.0D0 * APYY /  Y(2)
      FIVE  =  2.0D0 * APYY /  Y(2)
      SIX   = 12.0D0 * APYX /( Y(3)**3 )
      SEVEN = E(5) * PI * DIA(5)**2 /( Y(5)*4.0D0 )
      EIGHT =  6.0D0 * APYX /( Y(3)**2 )
      ANINE =  4.0D0 * APYX /  Y(3)
      TEN   = E(4) * PI * DIA(4)**2 /( Y(4)*4.0D0 )
      TEN = 254.0D0
      ELVEN =  2.0D0 * APYX /  Y(3)
C
      STF(1,1,C) = ONE
      STF(1,8,C) = -ONE
      STF(2,2,C) = TWO
      STF(2,3,C) = THREE
      STF(2,6,C) = THREE
      STF(3,3,C) = FOUR
      STF(3,6,C) = FIVE
      STF(4,4,C) = SIX + SEVEN + TEN
      STF(4,5,C) =-EIGHT
      STF(4,6,C) =-EIGHT
      STF(4,7,C) = SEVEN
      STF(5,5,C) = ANINE
      STF(5,6,C) = ELVEN
```

```
      STF(6,6,C) = ANINE + FOUR
      STF(7,7,C) = SEVEN
      STF(8,8,C) =  ONE
C
      SYSM(1,1,C) =  AM(1) / 3.0D0
      SYSM(1,8,C) =  AM(1) / 6.0D0
      SYSM(2,2,C) =   156.0D0 * AM(2) / 420.0D0
      SYSM(2,3,C) =  AM(2) * 22.0D0 * Y(2) / 420.0D0
      SYSM(2,6,C) = -AM(2) * 13.0D0 * Y(2) / 420.0D0
      SYSM(3,3,C) =  AM(2) *  4.0D0 * Y(2)**2 / 420.0D0
      SYSM(3,6,C) = -AM(2) *  3.0D0 * Y(2)**2 / 420.0D0
      SYSM(4,4,C) =  AM(3) *156.0D0 / 420.0D0 + ( AM(5) + AM(4) )/ 3.0D0
      SYSM(4,5,C) = -AM(3) * 22.0D0 * Y(3) / 420.0D0
      SYSM(4,6,C) =  AM(3) * 13.0D0 * Y(3) / 420.0D0
      SYSM(4,7,C) = -AM(5) / 6.0D0
      SYSM(5,5,C) =  AM(3) *  4.0D0 * Y(3)**2 / 420.0D0
      SYSM(5,6,C) = -AM(3) *  3.0D0 * Y(3)**2 / 420.0D0
      SYSM(6,6,C) = SYSM(3,3,C) + SYSM(5,5,C)
      SYSM(7,7,C) =  AM(5) / 3.0D0
      SYSM(8,8,C) =  AM(1) / 3.0D0
C
      IF(C .EQ. 1)  GO TO 12
      STF(1,2,3) = -ONE
      STF(2,2,3) = ONE + TWO
      SYSM(1,2,3) =  AM(1) / 6.0D0
      SYSM(2,2,3) =  AM(1) / 3.0D0 + 156.0D0*AM(2)/420.0D0
C
      ZTA(1) = 0.1D0
      ZTA(4) = 0.5D0
   12 CONTINUE
      DO 20 JA = 1, JC
C
   15 DO 20 JB = 1, JC
      STF(JB,JA,C)  =  STF(JA,JB,C)
      SYSM(JB,JA,C) = SYSM(JA,JB,C)
      DIAGC(JA,JB)  = SYSM(JA,JB,C)
      PHINV(JA,JB) =  STF(JA,JB,C)
   20 CONTINUE
C
   24 CALL DVVS( PHINV, DIAGC, ELAM(1,C), PHI(1,1,C), JC, JC, JC, JC,
     1            KSYS, NERR)
C
      IF( NERR .NE. 0)  WRITE(NUNIT,330) NERR
C
   25 DO 37  JA = 1, JC
      ELAM(JA,C) = ELAM(JA,C)**.5
   37 CONTINUE
C
```

```
      DO 1310  JA = 1,JC
C
   35 DO 1310  JB = 1,JC
      PHTINV(JA,JB) = PHI(JB,JA,C)
      PHINV(JA,JB) = PHI(JA,JB,C)
C
      DIAGC(JA,JB) = 0.0D0
 1310 CONTINUE
C
C
 1312 DO 1315  KA = 1,JC
C
 1314 DO 1315  KB = 1,JC
      PHITK(KA,KB) = 0.0D0
      PHITM(KA,KB) = 0.0D0
C
      DO 1315  KC = 1,JC
      PHITM(KA,KB) = PHTINV(KA,KC)*SYSM(KC,KB,C) + PHITM(KA,KB)
      PHITK(KA,KB) = PHTINV(KA,KC)* STF(KC,KB,C) + PHITK(KA,KB)
C
 1315 CONTINUE
C
      DO 1318  KA = 1,JC
C
      DO 1318  KB = 1,JC
      DIAGK(KA,KB) = 0.0D0
      DIAGM(KA,KB) = 0.0D0
C
 1316 DO 1318  KC = 1,JC
      DIAGM(KA,KB) = PHITM(KA,KC)*PHI(KC,KB,C) + DIAGM(KA,KB)
      DIAGK(KA,KB) = PHITK(KA,KC)*PHI(KC,KB,C) + DIAGK(KA,KB)
C
 1318 CONTINUE
C
      DO 1328  KA = 1,JC
      DIAGC(KA,KA) = 2.0D0*ZTA(KA)*( DIAGM(KA,KA)*DIAGK(KA,KA) )**0.5D0
 1328 CONTINUE
C
 1326 CALL DINVS(PHTINV,JC,KSYS)
C
```

```
C
 1327 CALL DINVS(PHINV,JC,KSYS)
C
C
      DO 1322  KA = 1,JC
C
      DO 1322  KB = 1,JC
      PHTIDC(KA,KB) = 0.0D0
C
 1323 DO 1322  KC = 1,JC
      PHTIDC(KA,KB) = PHTINV(KA,KC)*DIAGC(KC,KB) + PHTIDC(KA,KB)
C
 1322 CONTINUE
C
      DO 1324  KA = 1,JC
C
      DO 1324  KB = 1,JC
      SEE(KA,KB) = 0.0D0
C
1321  DO 1324  KC = 1,JC
      SEE(KA,KB) = PHTIDC(KA,KC)*PHINV(KC,KB) + SEE(KA,KB)
C
 1324 CONTINUE
C
      DO  1325  KA = 1, JC
 1330 DO  1325  KB = 1, JC
      DAMP(KA,KB,C) = SEE(KA,KB)*COFD
 1325 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE MONTOR(NCLEAR, CONFIG, ICORD, F,  C, NPRINT, IDT, DTP,
     1           DTPMIN, IMPACT, Q, QSAV, FSAV, DT)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER YES, C, CONFIG(16,1)
C
      DIMENSION ICORD(NCLEAR,1), F(1), FSAV(1),DTP(1),  Q(1), QSAV(1),
     1           IMPACT(1)
C
      COMMON NUNIT, IPRINT
C
      DATA YES/4HYES /, NO/4HNO  /
C
 5000 FORMAT('0', 20X, 'MONITORING INDICATES MORE THAN ONE CONFIGURATION
     1 CHANGE IN ONE TIME STEP')
 5001 FORMAT('0', 20X, '********** ERROR - - COMPUTER TRUNCATION - - KK
     1 NOT CORRECTLY SET **********')
 5002 FORMAT(' ',10X, 'DTP(', I2, ')=',1PD15.7, 15X, 'DTPMIN =', D15.7)
 5003 FORMAT(' ', 10X, 'INDEX OF DTP(I) FOR MIN IS', I2, ' AND IMPACT ='
     1,  I2)
C
      IDT = 0
      EPS1 = 1.0D-1
      EPS2 = 1.0D-6
C
      DO  20  I = 1, NCLEAR
      DTP(I) = +1.0D0
      IMPACT(I) = 0
   20 CONTINUE
C
      DO 23  KK = 1, NCLEAR
C
      INDEX = ICORD(KK,1)
      NDEX = ICORD(KK,2)
C
      IF( CONFIG(C,KK) .EQ. 0 )  GO TO 12
C
C              ******************************************
C              **                                      **
C              **             FORCE MONITOR            **
C              **                                      **
C              ******************************************
C
      IF(F(INDEX) .GE. 0.0) GO TO 23
C
```

```
      FSAVE = FSAV( INDEX )
      FSEND = F( INDEX )
C
      CALL DTPRIM( FSEND, KK, FSAVE, DT, DTP, NPRINT)
      IDT = IDT + 1
C
      GO TO 23
C
C                ******************************************
C                **                                      **
C                **        DISPLACEMENT MONITOR          **
C                **                                      **
C                ******************************************
C
   12 DISPL = Q( INDEX ) - Q( NDEX )
      IF(DISPL .GE. 0.0)  GO TO  23
      IF( DABS(DISPL) .LE. EPS2)    GO TO 23
C
      IMPACT(KK) = KK
      DOLD = QSAV( INDEX ) - QSAV( NDEX )
       IF(DOLD .LE. 0.0)  DOLD = 1.0D-10
C
      CALL DTPRIM( DISPL, KK, DOLD, DT, DTP, NPRINT)
      IDT = IDT + 1
C
   23 CONTINUE
C
      DO  25  I = 1, NCLEAR
      IF(DTP(I) .LT. 0.1D0)  GO TO 50
   25 CONTINUE
      IDT = -1
C
      RETURN
C
C
   50 DTPMIN = +1.0D6
      DO 60  JA = 1, NCLEAR
      DTPMIN = DMIN1( DTPMIN, DTP(JA) )
      IF(NPRINT .EQ. YES) WRITE(NUNIT,5002) JA, DTP(JA),DTPMIN
   60 CONTINUE
C
      IF(IDT .GT. 1   .AND.  NPRINT .EQ. YES)  WRITE(NUNIT,5000)
C
      DO  70  KK = 1, NCLEAR
      IF(DTPMIN .EQ. DTP(KK) )  GO TO 80
   70 CONTINUE
C
      WRITE(NUNIT,5001)
```

```
C
C         ****************************************************
C         **                                                **
C         **            CHANGE CONFIGURATION                **
C         **                                                **
C         ****************************************************
C
   80 CALL SHIFT( C, KK, CONFIG, NPRINT)
      IDT = KK
C
      IF(NPRINT .EQ. NO)  RETURN
      WRITE(NUNIT,5003) KK, IMPACT(KK)
C
      RETURN
      END
```

```
      SUBROUTINE CAM(WT,AMP,AMPW,XZERO,W2,Q,QD,QDD)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER YES
C
      DIMENSION Q(1), QD(1), QDD(1)
C
      COMMON NUNIT, IPRINT
C
      DATA  YES/4HYES /, NO/4HNO  /
C
      AMPSIN = AMP*DSIN(WT)
      Q(9)   = AMPSIN + XZERO
      QD(9)  = AMPW*DCOS(WT)
      QDD(9) = AMPSIN*W2
C
      RETURN
      END
```

```
      SUBROUTINE NEWMRK( ISYS, KSYS,     C,      STF , DAMP, SYSM, DP,
     1  DT, DT02, DT206, DT203, Q, QD, QDD, BMIV, DA, DV, NC,
     2  ALFA,BETA,CONFIG,F,ICORD,NCLEAR,NPRINT,IQ,COLM,COLC,COLK,IDP,
     3  PIN2$8, IDIM, LAYER, IGU, QDDSAV, QDSAV, QSAV, SCALE, IMP)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER C, CONFIG(16,1), YES, PIN2$8
C
      DIMENSION BMIV(KSYS,KSYS), DA(KSYS,KSYS), DV(KSYS,KSYS),
     1  STF(KSYS,KSYS,1), DAMP(KSYS,KSYS,1), SYSM(KSYS,KSYS,1),
     2  ALFA(1), BETA(1), Q(1), QD(1), QDD(1), DP(1),
     3  ICORD(NCLEAR,1), F(1), IQ(KSYS,1), COLM(KSYS,1), COLC(KSYS,1),
     4  COLK(KSYS,1), IDP(KSYS,1), IDIM(1), LAYER(1), QDDSAV(1),
     5  QDSAV(1), QSAV(1)
C
      COMMON NUNIT, IPRINT
C
      DATA  YES/4HYES /, NO/4HNO  /
C
 5000 FORMAT('0',10X,'ICORD(',I1,')=',I3)
 5007 FORMAT(' ',10X, 'SYSM(', I1, ',', I1, ')=',1PD15.8,4X,
     1               'DAMP(', I1, ',', I1, ')=',1PD15.8,4X,
     2               'STF(', I1, ',', I1, ')=',1PD15.8)
 5008 FORMAT('0','ALFA*',8(3X,1PD12.5) )
 5009 FORMAT(' ','BETA*',8(3X,1PD12.5) )
 5010 FORMAT(' ',1P7D18.6)
 5011 FORMAT(' ',9X,'BMIV',14X,'DP',16X,'DA',14X,'ALFA',14X,'DV',
     1  15X,'BETA',12X,'QDD')
 5015 FORMAT('0')
 5027 FORMAT('0',///,10X,'CONFIGURATION MATRIX' )
 5028 FORMAT('0',//,10X,2(4X,I1) )
 5037 FORMAT(' ','FORCE',1P7D16.8, D15.7)
 5041 FORMAT('0','******************** IN SUBROUTINE NEWMRK - -  C =',
     1     I2, '****************')
 5042 FORMAT(' ','******************** RETURNED TO MAIN  OR  BACKSTEP ',
     1     'FROM NEWMRK  *******',//)
 5043 FORMAT(' ', 20X, 'C =', I2, 10X, ' NCLEAR =',I2)
 5044 FORMAT(' ',30X, ' CONFIG ELEMENT - -',I2,',',I2)
 5045 FORMAT(' ', 10X, 'IMPACT ACCEL AT COORD 2 IS ', D15.7, ' BASED ON'
     2           ,F2.3, ' TIMES  THE OLD VELOCITY', 1PD15.7)
C
C
      IF(NPRINT .EQ. YES)  WRITE(NUNIT,5041) C
C
   25 DO  50  I = 1,KSYS
```

```
      DP(I) = 0.0D0
   50 CONTINUE
C
      IF(C .EQ. 1  .OR.  C .EQ. 3)  GO TO 500
C
      DO  300  JA = 1,NCLEAR
      KDP = IDP(C,JA)
      IF(KDP .NE. 0)  GO TO 325
C
  300 CONTINUE
C
  325 KC = C
C
  350 ACCEL  =  QDD(KDP)
      VEL    =   QD(KDP)
      DIST   =    Q(KDP)
C
      DO  400  JA = 1, ISYS
      CSYSM = COLM(JA,KC)
      CDAMP = COLC(JA,KC)
      CSTFF = COLK(JA,KC)
      DP(JA) = CSYSM*ACCEL + CDAMP*VEL + CSTFF*DIST + DP(JA)
  400 CONTINUE
C
  425 IF(KC .NE. C)  GO TO 500
      IF(C .EQ. 6)  KC = 1
      IF(C .EQ. 8)  KC = 3
      KDP = 7
      IF(KC .NE. C)  GO TO 350
  500 CONTINUE
C
  600 DO  700  JA = 1, ISYS
      IA = IQ(JA,C)
      ACCEL =  QDDSAV(IA)
      VEL   =   QDSAV(IA)
C
      ALFA(JA) = ACCEL*DT02  + VEL
      BETA(JA) = ACCEL*DT203 + VEL*DT + QSAV(IA)
  700 CONTINUE
C
      IF(NPRINT .EQ. NO)  GO TO 750
C
      WRITE(NUNIT,5008) ( ALFA(JA),  JA = 1,ISYS )
      WRITE(NUNIT,5009) ( BETA(JA),  JA = 1,ISYS )
      WRITE(NUNIT,5011)
C
  750 DO  900  JA = 1, ISYS
C
```

```
      IA = IQ(JA,C)
      QDD(IA) = 0.0D0
C
      DO  800  JB = 1, ISYS
      QDD(IA) = QDD(IA) - BMIV(JA,JB)*DP(JB) - DA(JA,JB)*ALFA(JB)
     1   - DV(JA,JB)*BETA(JB)
C
      IF(NPRINT .EQ. NO)  GO TO 800
      WRITE(NUNIT,5010) BMIV(JA,JB),DP(JB),DA(JA,JB),ALFA(JB),DV(JA,JB),
     1   BETA(JB), QDD(IA)
C
  800 CONTINUE
C
      IF(NPRINT .EQ. YES)  WRITE(NUNIT,5015)
  900 CONTINUE
C
      IF(IMP .NE. 2)  GO TO 950
      QDD(2) = (SCALE - 1.0D0) * QDSAV(2) / DT
      WRITE(NUNIT,5045) QDD(2), SCALE, QDSAV(2)
      IMP = -10
  950 DO  1000  JA = 1, ISYS
      IA = IQ(JA,C)
      QD(IA) = ALFA(JA) + QDD(IA)*DT02
      Q (IA) = BETA(JA) + QDD(IA)*DT2O6
C
 1000 CONTINUE
      IF(IMP .NE. -10) GO TO 1050
      WRITE(NUNIT,5046) QD(2)
 5046 FORMAT(' ', 10X, 'NEW CALCULATED VELOCITY OF COORD. TWO IS  ',
     1        1PD15.7)
      IMP = 0
 1050 CONTINUE
C
      LC = LAYER(C)
      IF(LC .EQ. 3)  GO TO 1170
 1150 IF(PIN2$8 .EQ. NO)  GO TO 1175
C
 1170 CALL ADJUST(8,2, Q, QD, QDD, Q, QD, QDD)
      LC = 3
 1175 CONTINUE
      IF(IGU .EQ. YES)  LC = 1
      ILC = IDIM(LC)
C
      DO  1300  IA = 1, NCLEAR
      KK = ICORD(IA,1)
      F( KK) = 0.0D0
C
      DO  1200  JA = 1, ILC
```

```
      F(KK) = F(KK ) + SYSM(KK,JA,LC)*QDD(JA)
     1   + DAMP(KK,JA,LC)*QD(JA) + STF(KK,JA,LC)*Q(JA)
 1200 CONTINUE
C
 1300 CONTINUE
 1350 IF(PIN2$8 .EQ. NO)  GO TO 1375
C
      F(2) = 0.0D0
      F(8) = 0.0D0
C
 1375 CONTINUE
      IF(NPRINT .EQ. NO)  RETURN
      WRITE(NUNIT,5037)    ( F(JQ), JQ = 1, KSYS )
C
      WRITE(NUNIT,5042)
C
      RETURN
      END
```

```
      SUBROUTINE  OLPLOT(X,Y,N,XMAX,XMIN,YMAX,YMIN,IRYP1, GRID,XSCALE,
     1                MODCUR, NPRINT, LINES)
C             ******************************************
C             **                                      **
C             **        THIS  SUBROUTINE              **
C             **     PLOTS A SINGLE PRECISION FUNCTION **
C             ** LENGTHWISE ON UP TO 6 PAGES OF OUTPUT **
C             **                                      **
C             ******************************************
      INTEGER*2 GRID,BLANK,DASH,SLINE,YCHAR(4)/1H.,1H+,1H*,1HX/
      DIMENSION GRID(LINES,1),YSCALE(5),XSCALE(1)
      DIMENSION Y(1),X(1)
      DATA DOT,BLANK,DASH,SLINE/Z4B40,Z4040,Z6D40,Z4F40/
      CCMMON NUNIT, IPPINT
      NDATA = N
      KKZ = 1
C
C     RY IS THE NUMBER OF LENGTHWISE LINES OF OUTPUT PLOT DESIRED
C
      RY = FLOAT(IRYP1 - 1 )
      IXSC = IRYP1/10 + 1
      IERR=0
C
C
C
C     GRID IS THE MATRIX USED TO PLOT THE POINTS
C
C     CHECKING Y AND X POINTS AND PLOTTING THOSE OUT OF RANGE
C     AT THE MARGIN
C
      DO 30 I=1,NDATA,KKZ
      IF(Y(I)-YMAX) 205,205,220
  220 Y  (I)=YMAX
      IERR=IERR+1
      GOTO 210
  205 IF(Y(I)-YMIN)203,210,210
  203 Y  (I)=YMIN
      IERR=IERR+1
  210 IF(X(I)-XMAX)215,215,212
  212 X  (I)=XMAX
      IERR=IERR+1
      GOTO 30
  215 IF(X(I)-XMIN)217,  30,30
  217 X  (I)=XMIN
      IERR=IERR+1
C
```

```
   30 CONTINUE
C
C        PLOTTING Y AND X AYIS , IF NECESSARY
C
         YRANGE=YMAX-YMIN
         XRANGE=XMAX-XMIN
         IF (XRANGE.NE.0.)  GO TO 298
         XMIN=YMIN
         XMAX=YMAX
         XRANGE=YRANGE
         GO TO 299
  298    IF (YRANGE.NE.0.)  GO TO 299
         YMIN=XMIN
         YMAX=XMAX
         YRANGE=XRANGE
C
C        BLANKING OUT MATRIX-(GRID)
C
  299    DO 300 I=1,IRYP1
         DO 301 JJ=1,81
  301    GRID(I,JJ)=BLANK
  300    CONTINUE
         XTEST=XMAX*XMIN
         YTEST=YMAX*YMIN
         IF(YTEST)1,222,222
  222    IF(XTEST)333,444,444
    1    IXAXIS=80. *(-YMIN)/YRANGE+1.5
         DO 40 I=1,IRYP1
   40    GRID(I,IXAXIS)=SLINE
         GOTO 222
  333    IYAXIS=RY *XMAX/XRANGE+1.5
         DO 60 I=1,81
   60    GRID(IYAXIS,I)=DASH
C
C        PLACING POINTS IN THEIR PROPER GRID POSITIONS
  444    IF(MODCUR.EQ.0.OR.MODCUR.EQ.1)JSET=0
         JSET=JSET+1
         IF(JSET.GT.4) JSET=1
         DO 700 I=1,NDATA,KKZ
         IPTY=RY *(X(I)-XMIN)/XRANGE+1.5
         IPTX=80. *(Y(I)-YMIN)/YRANGE+1.5
         IF(IPTY.GT.IRYP1.OR.IPTX.GT.81) GO TO 70
         IF(IPTY.LE.0.OR.IPTX.LE.0)GO TO 70
         GRID(IPTY,IPTX) = YCHAR(JSET)
         GO TO 700
   70    IERR=IERR+1
  700    CONTINUE
C
```

```
C     COMPUTE PROPER SCALE NUMBERS
C
      IF(MODCUR.EQ.0.OR.MODCUR.EQ.1) GO TO 8000
      IF(MODCUR.EQ.2) RETURN
      GO TO 922
 8000 YINCR=YRANGE/4.
      XINCR=XRANGE/FLOAT(IXSC-1)
      YSCALE(1)=YMAX
      XSCALE(1)=XMIN
      DO 80 I=2,5
   80 YSCALE(I)=YSCALE(I-1)-YINCR
      DO 81 I=2,IXSC
   81 XSCALE(I)=XSCALE(I-1)+XINCR
C
C     OUTPUT SECTION WITH GRAPH
C
      IF(MODCUR.EQ.0.OR.MODCUR.EQ.3)GO TO 922
      RETURN
   17 FORMAT(12X,
     1             1P E10.3,4(10X,E10.3)/15X,2H**,8(10H+*********),3H+**)
  922 WRITE(6,17) YSCALE(5),YSCALE(4),YSCALE(3),YSCALE(2),YSCALE(1)
      II=1
      I=0
      DO 101 IK=1,IRYP1
      IF(I)91,91,92
   91 WRITE(6,18) XSCALE(II),(GRID(IK,IY),IY=1,81),XSCALE(II)
 18   FORMAT(3X,1P
     1              E10.3,2X,1H+,1X,81A1,1X,1H+,2X,E10.3)
      II=II+1
      GOTO 102
   92 WRITE(6,19) (GRID(IK,IY),IY=1,81)
   19 FORMAT(15X,1H*,1X,81A1,1X,1H*)
  102 I=I+1
      IF(I-10)101,103,103
  103 I=0
  101 CONTINUE
      WRITE(6,22) YSCALE(5),YSCALE(4),YSCALE(3),YSCALE(2),YSCALE(1)
   22 FORMAT(15X,2H**,8(10H+*********),3H+**/   1P
     1                                           12X,E10.3,4(10X,E10.3))
      IF(IERR) 1000,1000,1001
 1001 WRITE(6,20) IERR
 20   FORMAT(10X 'NUMBER OF POINTS OUT OF RANGE ='  I4)
C
 1000 RETURN
      END
```

```
      SUBROUTINE PRINT(KSYS, ISYS, C, SYSM, DAMP, STF, BMIV, DA, DV,
     1          CONFIG, PHI, ELAM, STFI, STFRI, WTSTOP, WTPRNT, LABEL,
     2          YNAME, XNAME, TITLE,
     3          AGTOUT, CARDS, TAPE, IPLOT, IDRAW, DAMPNG, RPM, CLEAR,
     4          AMP, DELT, ROCKER, AXILTH, REALTH, NTIME, NC,
     5          NCLEAR, KSYSM1, COLM, COLC, COLK, COFD, ICORD,
     6          LAYER, IDP, IQ, IDIM, KSYSM2, PIN2$8, XZERO, FZERO,
     7          MPRINT)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      REAL*4 LABEL,TITLE,XNAME,YNAME
C
      INTEGER YES,AGTOUT,DAMPNG,TAPE,CARDS,SEAT,C,CONFIG(16,1),PIN2$8
C
      DIMENSION PHI(KSYS,KSYS), ELAM(KSYS), SYSM(KSYS,KSYS),
     1          DAMP(KSYS,KSYS),   STF(KSYS,KSYS),   BMIV(KSYS,KSYS),
     2          DA(KSYS,KSYS),   DV(KSYS,KSYS),
     3          STFI(KSYSM1,1), STFRI(KSYSM2,1), LABEL(1), TITLE(1),
     4          YNAME(20,1), AXILTH(1), REALTH(1),
     5          XNAME(1), COLM(KSYS,1), COLC(KSYS,1), COLK(KSYS,1),
     6          ICORD(NCLEAR,1)
      DIMENSION LAYER(1), IDP(  KSYS,1), IQ(KSYS,1), IDIM(1)
C
      COMMON NUNIT, IPRINT
C
      DATA  YES/4HYES /, NO/4HNO  /
C
  620 FORMAT('0',10X,' MODE SHAPES-(PHI)',/)
  660 FORMAT('0',10X,' NATURAL FREQ-(WN)',/)
 1020 FORMAT('0',50X,'- ( BMIV ) -' ,//)
 1030 FORMAT(' ',20X, 1P8D10.2)
 1040 FORMAT('0',50X,'- ( DA ) -' ,//)
 1049 FORMAT('1', //////)
 1050 FORMAT('0', 40X, ' CONFIGURATION AND MATRIX LAYER =', I3,//,50X,
     1   'MATRICES ARE',I3,'X',I1,//)
 1060 FORMAT('0',50X,'- ( DV ) -' ,//)
 1070 FORMAT('0',//, 50X, '--- SYSM --- ',//)
 1071 FORMAT('0',//,50X, ' --- DAMP ---',//)
 1072 FORMAT('0',//,50X, ' --- STF ---',//)
 5000 FORMAT('0', 2(10X,'ICORD(',I1,',' ,I1,  ')=',I3) )
 5001 FORMAT('1', ////////)
```

```
5002 FORMAT('0')
5003 FORMAT('0', 20X, ' COORDINATES 2 AND 8 ARE PINED - - TWO ',
    1          'CLEARANCE PROBLEM')
5004 FORMAT('0', 20X, ' COORDINATES 2 AND 8 ARE NOT PINED - - THREE ',
    1          'CLEARANCE PROBLEM')
5010 FORMAT('0',//,20X,5HRPM =,0PF9.2,
    1     5X15HDAMPING COEF. =,0PF6.3,5X,7HCLEAR =,0PF6.3)
5017 FORMAT('0', 20X, ' XZERO=', 1PD15.7,   '      FZERO =', D15.7)
5020 FORMAT('0',20X,' AGT OUTPUT REQUESTED')
5021 FORMAT('0',20X,' NO AGT OUTPUT REQUESTED')
5022 FORMAT('0',20X,' AGT OUTPUT ON MAGNETIC TAPE')
5023 FORMAT('0',20X,' AGT OUTPUT ON PUNCHED CARDS')
5024 FORMAT('0',20X,' DAMPING REQUESTED')
5025 FORMAT('0',20X,'NO DAMPING REQUESTED')
5026 FORMAT('0',20X,18A4)
5027 FORMAT('1',///,10X,'CONFIGURATION MATRIX' )
5028 FORMAT('0',//,10X,3(4X,I1) )
5041 FORMAT('0',20X,' VALVE SEAT IMPOSED WITH AN INITIAL CLEARANCE OF '
    1        , F5.2, ' INCHES' )
5042 FORMAT('0', 20X, 'NO VALVE SEAT')
5043 FORMAT(' ',//,10X,'COL-M-C-K ',//,6(10X,1P3D18.9,//)  )
5044 FORMAT(' ',  1P7D18.9)
5045 FORMAT('0',//,20X,'- - - - -  STFI - - - - - - ',//)
5046 FORMAT('0',//,20X,'- - - -  STFRI - - - - -  ', //)
5047 FORMAT('0',20X,' CALCOMP GRAPH DESIRED ')
5048 FORMAT('0',20X,' CALCOMP GRAPH NOT DESIRED ')
5050 FORMAT('0', 30X,  'MATRIX  - LAYER -' )
5051 FORMAT('0', 35X, I1)
5052 FORMAT('0', 30X,  'MATRIX  - IDP -' )
5053 FORMAT('0', 30X, 3( I2, 2X) )
5054 FORMAT('1',///, 30X,  'MATRIX  - IQ -' )
5055 FORMAT('0', 25X, 8( I1, 2X) )
5056 FORMAT('0', 30X,  'MATRIX  - IDIM -' )
5057 FORMAT('0', 35X, I1 )
5058 FORMAT('0', ///, 30X, ' MATRIX - ICORD -')
5312 FORMAT('0', 5X, 'WTSTOP =',1PD18.8, 5X, 'WTPRNT =',D18.8)
C
C
      IF(C .EQ. 3)  GO TO 358
      IF(C .EQ. 2  .OR.  C .GE. 4)  GO TO 359
C
      WRITE(NUNIT,5001)
C
```

```
C
      WRITE(NUNIT,5026) (  LABEL(JA), JA = 1,18 )
      WRITE(NUNIT,5026) (  TITLE(JA), JA = 1,18 )
      WRITE(NUNIT,5026) ( YNAME(JA,1 ),JA = 1,18 )
      WRITE(NUNIT,5026) ( YNAME(JA,2 ),JA = 1,18 )
      WRITE(NUNIT,5026) ( YNAME(JA,3 ),JA = 1,18 )
      WRITE(NUNIT,5026) ( YNAME(JA,4 ),JA = 1,18 )
      WRITE(NUNIT,5026) ( YNAME(JA,5 ),JA = 1,18 )
      WRITE(NUNIT,5026) ( YNAME(JA,6 ),JA = 1,18 )
      WRITE(NUNIT,5026) ( YNAME(JA,7 ),JA = 1,18 )
      WRITE(NUNIT,5026) ( YNAME(JA,8 ),JA = 1,18 )
      WRITE(NUNIT,5026) ( YNAME(JA,9 ),JA = 1,18 )
      WRITE(NUNIT,5026) ( YNAME(JA,10),JA = 1,18 )
      WRITE(NUNIT,5026) ( YNAME(JA,11),JA = 1,18 )
      WRITE(NUNIT,5026) ( YNAME(JA,12),JA = 1,18 )
      WRITE(NUNIT,5026) ( YNAME(JA,13),JA = 1,18 )
C
      IF( AGTOUT .EQ. YES )  WRITE(NUNIT,5020)
      IF( AGTOUT .EQ. NO  )  WRITE(NUNIT,5021)
C
      IF( TAPE .EQ. YES )   WRITE(NUNIT,5022)
C
      IF( CARDS .EQ. YES )   WRITE(NUNIT,5023)
C
      IF( DAMPNG .EQ. NO )  WRITE(NUNIT,5025)
      IF( DAMPNG .EQ. YES)  WRITE(NUNIT,5024)
C
      IF(IDRAW .EQ. YES)  WRITE(NUNIT,5047)
      IF(IDRAW .EQ. NO )  WRITE(NUNIT,5048)
C
      IF(CLEAR .LE. -10.0)  WRITE(NUNIT,5042)
      IF(CLEAR .GT. -10.0)  WRITE(NUNIT,5041)  CLEAR
C
      IF(PIN2$8 .EQ. YES)  WRITE(NUNIT,5003)
      IF(PIN2$8 .EQ.  NO)  WRITE(NUNIT,5004)
C
C
      WRITE(NUNIT,5010)      RPM,COFD , CLEAR
      WRITE(NUNIT,5017)  XZERO, FZERO
      IF(MPRINT .EQ. NO)  RETURN
C
      WRITE(NUNIT,5027)
      DO  80  JA = 1, NC
      WRITE(NUNIT,5028)  (CONFIG(JA,JB), JB = 1,NCLEAR)
```

```
   80 CONTINUE
C
      WRITE(NUNIT,5050)
      DO  700  JA = 1, KSYS
      WRITE(NUNIT,5051)  LAYER(JA)
  700 CONTINUE
C
      WRITE(NUNIT,5002)
      WRITE(NUNIT,5052)
      DO  710  JA = 1, KSYS
      WRITE(NUNIT,5053)  (IDP(JA,JB), JB = 1, NCLEAR)
  710 CONTINUE
C
      WRITE(NUNIT,5054)
      DO  720  JA = 1, KSYS
      WRITE(NUNIT,5055)  (IQ(JA,JB), JB = 1, KSYS)
  720 CONTINUE
C
      WRITE(NUNIT,5002)
      WRITE(NUNIT,5056)
      DO  730  JA = 1, KSYS
      WRITE(NUNIT,5057)  IDIM(JA)
  730 CONTINUE
C
      WRITE(NUNIT,5058)
      DO  740  JA = 1, NCLEAR
      WRITE(NUNIT,5053) (ICORD(JA,JB), JB = 1,2)
  740 CONTINUE
C
      WRITE(NUNIT,5001)
C
      WRITE(NUNIT,5045)
C
      DO  221  JA = 1,KSYSM1
      WRITE(NUNIT,5044)  (STFI(JA,JB), JB = 1,KSYSM1)
  221 CONTINUE
C
      WRITE(NUNIT,5046)
C
      DO  7  JA = 1,KSYSM2
      WRITE(NUNIT,5044)  (STFRI(JA,JB), JB = 1,KSYSM2)
    7 CONTINUE
C
  358 CONTINUE
C
      WRITE(NUNIT,620)
C
      DO 640  JA = 1,ISYS
```

```
      WRITE(NUNIT,1030) ( PHI(JA,JB), JB = 1,ISYS)
  640 CONTINUE
C
      WRITE(NUNIT,660)
      WRITE(NUNIT,1030) ( ELAM(JA), JA = 1,ISYS)
C
C
  359 CONTINUE
C
      IF(MPRINT .EQ. NO)  RETURN
C
      WRITE(6,1049)
    2 WRITE(NUNIT,1050) C,ISYS,ISYS
C
C
      WRITE(NUNIT,1020)
      DO 40  JA = 1, ISYS
   40 WRITE(NUNIT,1030) ( BMIV(JA,JB), JB = 1, ISYS)
C
      WRITE(NUNIT,5002)
      WRITE(NUNIT,1040)
      DO 50  JA = 1, ISYS
      WRITE(NUNIT,1030) ( DA(JA,JB), JB = 1, ISYS)
   50 CONTINUE
C
      WRITE(NUNIT,5002)
      WRITE(NUNIT,1060)
      DO 60  JA = 1, ISYS
      WRITE(NUNIT,1030) ( DV(JA,JB), JB = 1, ISYS)
   60 CONTINUE
C
   70 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE PRNTMX(COLM, COLC, COLK, SYSM, DAMP, STF, KSYS, C,
     1                  ISYS)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER C
C
      DIMENSION COLM(KSYS),   COLC(KSYS),   COLK(KSYS),
     1          DAMP(KSYS,KSYS), STF(KSYS,KSYS), SYSM(KSYS,KSYS)
C
      COMMON NUNIT, IPRINT
C
      DATA  YES/4HYES /, NO/4HNO  /
C
 1030 FORMAT(' ',20X, 1P8D10.2)
 1031 FORMAT('0', 27X, 1PD10.2)
 1032 FORMAT('+', 47X, 1PD10.2)
 1033 FORMAT('+', 72X, 1PD10.2)
 1048 FORMAT('0',//////)
 1049 FORMAT('1', //////)
 1050 FORMAT('0', 40X, ' CONFIGURATION AND MATRIX LAYER =', I3,//,50X,
     1   'MATRICES ARE',I3,'X',I1,//)
 1051 FORMAT('0', 40X, ' CONFIGURATION AND MATRIX LAYER =', I3,//,50X,
     1   'MATRICES ARE',I3,'X1',//)
 1070 FORMAT('0',//, 50X, '--- SYSM --- ',//)
 1071 FORMAT('0',//,50X, ' --- DAMP ---',//)
 1072 FORMAT('0',//,50X, ' --- STF ---',//)
 1073 FORMAT('0', ///, 40X, 'COLUMN MATRICES FOR CALCULATION OF DELTA P'
     1           ,//, 32X, 'COLUMN MASS',  8X, 'COLUMN DAMPING',
     2                10X, 'COLUMN STIFFNESS' )
C
C             ****************************************
C             **                                    **
C             **           PRINT  SYSTEM            **
C             **  MASS - DAMPING - STIFFNESS MATRICES   **
C             **      AND PRECALCULATED  MATRICES   **
C             **          COLM - COLC - COLK        **
C             **                                    **
C             ****************************************
C
      WRITE(6,1049)
    2 WRITE(NUNIT,1050) C,ISYS,ISYS
      WRITE(NUNIT,1070)
```

```
C
      DO  3  JA = 1,ISYS
      WRITE(NUNIT,1030) ( SYSM(JA,JB) ,  JB = 1,ISYS)
    3 CONTINUE
C
      WRITE(NUNIT,1071)
      DO  4  JA = 1,ISYS
      WRITE(NUNIT,1030) ( DAMP(JA,JB) ,  JB = 1,ISYS)
    4 CONTINUE
C
      WRITE(NUNIT,1072)
      DO  5  JA = 1,ISYS
      WRITE(NUNIT,1030) ( STF(JA,JB) ,  JB = 1,ISYS)
    5 CONTINUE
C
      WRITE(6,1049)
      WRITE(6,1048)
      WRITE(NUNIT,1051) C,ISYS
      WRITE(NUNIT,1073)
C
      DO  6  JA = 1, ISYS
      WRITE(NUNIT,1031)  COLM(JA)
      WRITE(NUNIT,1032)  COLC(JA)
      WRITE(NUNIT,1033)  COLK(JA)
    6 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE  REDUCE( SYSM, DAMP, STF, KSYS)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      DIMENSION  SYSM(KSYS,KSYS,1), DAMP(KSYS,KSYS,1), STF(KSYS,KSYS,1)
C
      DO  10  JA =1,KSYS
      DO  10  JB =1,KSYS
      SYSM(JA,JB,3) = SYSM(JA,JB,1)
      DAMP(JA,JB,3) = DAMP(JA,JB,1)
      STF (JA,JB,3) = STF (JA,JB,1)
   10 CONTINUE
C
      DO  20  JC = 1,KSYS
      SYSM(2,JC,3) = SYSM(2,JC,3) + SYSM(8,JC,3)
      DAMP(2,JC,3) = DAMP(2,JC,3) + DAMP(8,JC,3)
      STF (2,JC,3) = STF (2,JC,3) + STF (8,JC,3)
   20 CONTINUE
C
      DO  30  JD = 1,KSYS
      SYSM(JD,2,3) = SYSM(JD,2,3) + SYSM(JD,8,3)
      DAMP(JD,2,3) = DAMP(JD,2,3) + DAMP(JD,8,3)
      STF (JD,2,3) = STF (JD,2,3) + STF (JD,8,3)
   30 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE SHIFT( C, J, CONFIG, NPRINT)
C
      INTEGER CONSAV, CSAVE, C, CONFIG, BINARY, YES
C
      DIMENSION CONFIG(16,1)
C
      CCMMON NUNIT, IPRINT
C
      DATA  YES/4HYES /, NO/4HNO  /
C
      BINARY( I3,I2,I1) = I3*4 + I2*2 + I1
C
   10 FORMAT('0', 10X, 'FROM SHIFT, C=', I1,'  CONFIG(', I1,',', I1,
     1      ')=',I1)
   11 FORMAT(' ', 10X, 'NEW C = ',I1)
C
      IF(NPRINT .EQ. NO)  GO TO 12
C
      WRITE(NUNIT,10) C,C,J,CONFIG(C,J)
C
   12 CONSAV = CONFIG( C, J)
      CSAVE = C
      CONFIG(C,J) = CONSAV + 1
C
      IF( CONFIG(C,J) .EQ. 2 )  CONFIG(C,J) = 0
C
      C = BINARY( CONFIG(C,1), CONFIG(C,2), CONFIG(C,3) ) + 1
C
      IF(NPRINT .EQ. NO)   GO TO 20
C
      WRITE(NUNIT,11)  C
C
   20 CONFIG(CSAVE,J) = CONSAV
C
C
      RETURN
      END
```

```
      SUBROUTINE THESIS(FPLOT, DEG, DEG1, DEG2, NTIME, LINES, IP, GRID,
     1              Y1,Y2, Y3, Y4, YNAME, YNAME1, YNAME2, IGU, SCALE)
C
C
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      REAL*4 LABEL,TITLE,XNAME,YNAME1,YNAME2,YNAME,FPLOT,Y1,Y2,Y3,Y4,
     1      DEG, DEG1, DEG2, XSCALE(49)
      REAL*4 R4X, R4Y, QR4, WTR4, WNR4, RXZERO, RAMP, RRPM, RCLEAR
C
      INTEGER*2 GRID(LINES, 1)
C
      INTEGER YES,AGTOUT,DAMPNG,TAPE,CARDS,SEAT,C, PIN2$8
      INTEGER CONFIG(16,4), CSAVE, TWON, TWONP1
C
      DIMENSION FPLOT(NTIME,1),  Y1(NTIME,1), Y2(NTIME,1), Y3(NTIME,1),
     1          Y4(NTIME,1), YNAME(20,1), YNAME1(20,1), YNAME2(20,1),
     2          DEG(1), DEG1(NTIME), DEG2(NTIME,1)
      DIMENSION COLM(8,8), COLC(8,8), COLK(8,8), IDP(8,3), IQ(8,8)
      DIMENSION PHI(8,8,3), ELAM(8,3), IMPACT(3)
      DIMENSION SYSM(8,8,8),STF(8,8,8),DAMP(8,8,8),
     1          BMIV(8,8,9),  STFI(7,7),
     2          DA(8,8,9),DV(8,8,9),Q(10),QD(10),QDD(10),
     3          ALFA(8),BETA(8),STFRI(6,6),ICORD(3,2),
     4          LABEL(20),TITLE(20),XNAME(20),DP(8),
     7          QR4(10) , F(10), REALTH(4), AXILTH(4)
      DIMENSION QSAV(10), QDSAV(10), QDDSAV(10), FSAV(10),
     1          DTP(3)
      DIMENSION LAYER(8), IDIM(8)
C
C
      COMMON NUNIT, IPRINT
C
      DATA CONFIG/64*0/
      DATA YES/4HYES /, NO/4HNO  /
      DATA COLM/64*0.0D0/, COLC/64*0.0D0/, COLK/64*0.0D0/, IDP/24*0/
      DATA IQ/1,2,3,4,5,6,7,8, 1,2,3,4,5,6,8,0, 1,2,3,4,5,6,7,0,
     1     1,2,3,4,5,6,0,0, 2,3,4,5,6,7,8,0, 2,3,4,5,6,8,0,0,
     2     2,3,4,5,6,7,0,0, 2,3,4,5,6,0,0,0/
      DATA LAYER/2*1, 2*3, 2*1, 2*3/
      DATA FSAV/8*0.0D0/
      DATA LABEL(19)/'    '/, LABEL(20)/'    '/
      DATA TITLE(19)/'    '/, TITLE(20)/'    '/
      DATA XNAME(19)/'    '/, XNAME(20)/'    '/
C
```

```
C
 5000 FORMAT('0','CONFIGURATION =',I3,5X,'DEGREES =',F9.3,10X,'WT =',
     1        1PD15.7)
 5001 FORMAT('1')
 5002 FORMAT('0', 10X, 'BACKSTEP ROUTINE CALLED DUE TO CHANGE IN ',
     1        ' CONFIGURATIONS')
 5003 FORMAT(' ', 10X, 'RETURNED TO MAIN FROM BACKSTEP ROUTINE')
 5004 FORMAT(' ', 10X, 'IN MAIN - LOCATION ',I4,' - - -    CSAVE=', I2,
     1        '  AND C =', I2)
 5005 FORMAT('0','      W=', 1PD15.7,'    AMPW=', D15.7,'    W2=', D15.7,
     1        '   AMPW2=', D15.7)
 5007 FORMAT('0', 10X, ' VALUES CALCULATED AFTER BACKSTEP TAKEN:')
 5015 FORMAT(5E10.4,2A4)
 5016 FORMAT('0', 10X, 'IMPACT FORCE AT COORDINATE ',I2, '=',1PD15.7,
     1        '- - - DEGREES ', 0PF9.3)
 5026 FORMAT('0',10X,20A4)
 5038 FORMAT(20A4)
 5039 FORMAT('0',20X,20A4)
 5040 FORMAT('0',30X,20A4)
 5041 FORMAT(' ',5X,I5,5(10X,1PE10.3) )
 5050 FORMAT('0',40X,20A4)
 5051 FORMAT('0',50X,20A4)
 5062 FORMAT('0',23X,'DEGREES =',F9.3,10X,'WT =',1PD15.7)
 5063 FORMAT(' ','CAMQ =',1PD18.8, 10X,'CAMQD =', D18.8, 10X,'CAMQDD =',
     1     D18.8)
 5310 FORMAT('0',5(1PD15.7, 5X) )
C
C
C
C                ****************************************
C                **                                    **
C                **          READ INPUT VARIABLES      **
C                **                                    **
C                ****************************************
C
   50 CALL INPUT(LABEL, TITLE, XNAME, YNAME,
     1           AGTOUT, CARDS, TAPE, IPLOT, IDRAW,
     2           KPRINT, DAMPNG, RPM, CLEAR, WTDEG, AMP, DELT,
     3           ROCKER, AXILTH,
     4           REALTH, NTIME, KSYS, NCLEAR, NC,
     5           NCOUNT, IDEG, PIN2$8, MPRINT)
C
      FS = +145.0D0
      PI = 3.14159265359ODO
      DEGR = 0.
      WT = 0.
      NDEG = 0
      NDEGKP = 0
```

```
      ICOUNT = 0
      KAGT = 0
      IAGT = 1
      IDT = 0
      IMP = 0
      DTPM = 0.0D0
      COFD = 1.0D0
      IUNIT = 6
      ICORD(1,1) = 1
      ICORD(1,2) = 9
      ICORD(2,1) = 2
      ICORD(2,2) = 8
      ICORD(3,1) = 7
      ICORD(3,2) = 10
      DT = DELT
      NPRINT = IPRINT
      SEAT = YES
C
      NSYS = KSYS + 2
      DEGOPI = 180.0D0/PI
      KSYSM1 = KSYS-1
      KSYSM2 = KSYS-2
      KSYSM3 = KSYS - 3
      NCONF = NC
      NCP1 = NC + 1
C
      Q(10)      = CLEAR
      QD(10)     = 0.0D0
      QDD(10)    = 0.0D0
      QSAV(10)   = CLEAR
      QDSAV(10)  = 0.0D0
      QDDSAV(10)= 0.0D0
      W = RPM*PI/30.0D0
      AMPW = AMP*W
      W2 = -W*W
      AMPW2 = -AMPW*W
      DTW = DT*W
      WTSTOP = WTDEG   +15.0D0*DTW*DEGOPI
      WTPRNT = WTDEG   - 3.0D0*DTW*DEGOPI
C
      WRITE(NUNIT,5005) W, AMPW, W2, AMPW2
      WRITE(NUNIT,5310) WTSTOP,WTPRNT,WTDEG, DTW, DEGOPI
      WRITE(NUNIT,5039) (LABEL(JA),JA=1,18)
      WRITE(NUNIT,5039) (TITLE(JA),JA=1,18)
C
      IF( TAPE  .EQ. YES)  IUNIT  =  4
      IF( CARDS .EQ. YES)  IUNIT  =  7
      IF(CLEAR .LE. -10.0)  SEAT = NO
```

```
      IF(DAMPNG .EQ. NO)  COFD = 0.0D0
C
C              ****************************************
C              **                                    **
C              **          CALL  MCK                 **
C              **   SUBROUTINE MCK SETS UP THE SYSTEM **
C              **   MASS, DAMPING, AND STIFFNESS MATRICS **
C              **                                    **
C              ****************************************
C
      NE = 5
   75 CALL MCK(SYSM,DAMP,STF,KSYS,8,NE,NPRINT,COFD,PHI,ELAM,1)
      IF(IGU .EQ. YES)  GO TO 78
C
      CALL MCK(SYSM,DAMP,STF,KSYS,7,NE,NPRINT,COFD,PHI,ELAM,3)
C
      GO TO 80
   78 CALL REDUCE(SYSM,DAMP,STF,KSYS)
C
      DO 79  I = 1,KSYS
      ELAM(I,3) = ELAM(I,1)
   79 CONTINUE
C
C              ****************************************
C              **                                    **
C              **      ESTABLISH CONFIGURATION       **
C              **               MATRIX               **
C              **                                    **
C              ****************************************
C
   80 CALL CONMAT(NC, CONFIG, IDP, NPRINT, KSYS)
C
C              ****************************************
C              **                                    **
C              **      ESTABLISH REDUCED MATRICS     **
C              **                                    **
C              ****************************************
C
      DO  100  C = 1,KSYS
      CALL MATRIX(COLM, COLC, COLK, SYSM, DAMP, STF, STFI, KSYS,
     1            KSYSM1, C, IQ, MPRINT, LAYER, IDIM)
  100 CONTINUE
C
C
  110 XZERO = FS*ROCKER*STFI(1,1)+AMP
      FZERO = XZERO/STFI(1,1)
      WN = ELAM(1,3)
      IF(IGU .EQ. YES)  WN = ELAM(4,1)
```

```
C
C
C           *********************************************
C           **                                         **
C           **  COMPUTE INITIAL PROGRAM PARAMETERS FOR **
C           **         CONTACT AT CAM AND ROCKER       **
C           **                                         **
C           *********************************************
C
  120 CALL INITAL(KSYS,Q,QD,QDD,AMPW,FS,FZERO,XZERO,ROCKER,KSYSM2,
     1              COLM(1,7),COLC(1,7),COLK(1,7),DP,F,DAMP(1,1,7),
     2              STF(1,1,7), STFRI, NPRINT)
C
C           *********************************************
C           **                                         **
C           **          CALL SUBROUTINE DEL            **
C           **  DEL ESTABLISHES MATRICS USED IN THE    **
C           **  INTEGRATION ROUTINE. - - DEL SETS :    **
C           **     DELT/2,   DELT**2/3,   DELT**2/6    **
C           **               BMIVR, DAV, DVR           **
C           **                                         **
C           *********************************************
C
C
      DO  180  C = 1,NCONF
C
C
      ISYS = IDIM(C)
C
C
  183 CALL DEL(DT,DTO2,DT203,DT206,BMIV(1,1,C),DA(1,1,C),DV(1,1,C),
     1   SYSM(1,1,C), DAMP(1,1,C), STF(1,1,C), ISYS, C, NPRINT, KSYS,
     2   KPRINT)
C
C
      IF(MPRINT .EQ. NO  .AND.  C .NE. 1)  GO TO 180
      IC = 1
      IF(C .EQ. 3  .AND.  IGU .EQ. NO)   IC = 3
          CALL PRINT(KSYS, ISYS, C, SYSM(1,1,C), DAMP(1,1,C),
     2          STF(1,1,C), BMIV(1,1,C), DA(1,1,C), DV(1,1,C), CONFIG
     2          (1,2), PHI(1,1,IC), ELAM(1,IC), STFI, STFRI, WTSTOP,
     3          WTPRNT, LABEL, YNAME, XNAME, TITLE,
     4          AGTOUT, CARDS, TAPE, IPLOT, IDRAW, DAMPNG, RPM, CLEAR,
     5          AMP, DELT, ROCKER, AXILTH, REALTH, NTIME, NC,
     6          NCLEAR, KSYSM1, COLM, COLC, COLK, COFD, ICORD,
     7          LAYER, IDP, IQ, IDIM, KSYSM2, PIN2$8, XZERO, FZERO,
     8          MPRINT)
C
```

```
C
C
  180 CONTINUE
C
      WRITE(NUNIT,5001)
C
C     INITIAL CONTACT AT CAM AND ROCKER
  190 C = 7
      CSAVE = C + 1
C
C                  ********************************************
C                  **                                        **
C                  **          CONVERT TO REAL * 4           **
C                  **          OUTPUT FOR AGT                **
C                  **                                        **
C                  ********************************************
C
      IF( AGTOUT .EQ. NO ) GO TO  265
      RXZERO  =  XZERO
      RAMP  =  AMP
      RRPM  =  RPM
      RCLEAR = CLEAR
      WNR4 = WN
C
      IF(IUNIT .EQ. 6)  GO TO 260
      WRITE(IUNIT, 5038) (LABEL(JA),JA=1,18)
      WRITE(IUNIT, 5038) (TITLE(JA),JA=1,18)
C
  260 WRITE(IUNIT,5015) RXZERO,RAMP,RRPM, WNR4, RCLEAR, SEAT, PIN2$8
  265 CONTINUE
C
C-----------------------------------------------------------------------
C************************* CONFIGURATION CALCULATIONS ****************
C-----------------------------------------------------------------------
C
C                  ********************************************
C                  **                                        **
C                  **          INITIALIZING ROUTINE          **
C                  **                                        **
C                  ********************************************
C
  360 WT = WT + DTW
C
      DO 361  JA = 1, NSYS
      CALL ADJUST(JA,JA,QSAV,QDSAV,QDDSAV,Q,QD,QDD)
  361 CONTINUE
C
      DO 364  I = 1, KSYS
```

```
      FSAV(I) = F(I)
      FSAV(I) = DABS(F(I))
  364 CONTINUE
C
      IF(DEGR .GT.WTPRNT)   NPRINT = YES
      IF(DEGR .GT.WTSTOP)   STOP
C
  365 CALL CAM(WT,AMP,AMPW,XZERO,W2,Q,QD,QDD)
C
      IF(C .NE. CSAVE .AND. NPRINT .EQ. YES) WRITE(NUNIT,5000) C,DEGR,WT
      IF(C .EQ. CSAVE .AND. NPRINT .EQ. YES)  WRITE(NUNIT,5062) DEGR,WT
C
C
      CALL BRANCH(C,F,KSYS,ISYS,Q,QD,QDD,QDSAV,ICORD,DT,NPRINT,
     1          NCLEAR, IMP, PIN2$8, IDIM)
C
C
  390 CALL NEWMRK( ISYS, KSYS,     C,     STF , DAMP, SYSM, DP,
     1  DT, DTO2, DT206, DT203, Q, QD, QDD, BMIV(1,1,C), DA(1,1,C),
     2  DV(1,1,C),NC,ALFA,BETA,CONFIG(1,2),F,ICORD,NCLEAR,NPRINT,IQ,
     3  COLM, COLC, COLK, IDP, PIN2$8, IDIM, LAYER, IGU, QDDSAV,
     4  QDSAV, QSAV, SCALE, IMP)
C
      IF(((KPRINT .EQ. YES) .OR. (NPRINT .EQ. YES)).AND. (IMP .NE. 0))
     1            WRITE(NUNIT,5016) IMP, F(IMP), DEGR
C
C
      IF(NPRINT .EQ. YES)  CALL DEBUG( 2, Q, QD, QDD, F, KSYS, ISYS, DP,
     1                            NDEG, NPRINT, DEGR, C)
C
C
C
  400 CSAVE = C
  405 IF(NDEG .LT. IDEG)  GO TO 463
C
C              *********************************************
C              **                                         **
C              **            CHECK TO SEE IF              **
C              **       MONITORED VALUES  HAVE BEEN       **
C              **    EXCEEDED AND IF SO SHIFT TO THE      **
C              **       APPROPRIATE CONFIGURATION         **
C              **                                         **
C              *********************************************
C
  410 CALL MONTOR(NCLEAR, CONFIG(1,2), ICORD, F, C, NPRINT, IDT, DTP,
     1          DTPM, IMPACT, Q, QSAV, FSAV, DT)
C
```

```
C                 ********************************************
C                 **                                        **
C                 **  DETERMINE IF IMPACT AND AT WHAT COORD. **
C                 **                                        **
C                 ********************************************
C
      DO 415  IMP = 1,NCLEAR
      IF(IDT .EQ. IMPACT(IMP) )  GO TO 418
  415 CONTINUE
      IMP = 0
      GO TO 420
C
  418 IMP = ICORD(IMP,1)
  420 IF(CSAVE .EQ. C )  GO TO 450
C
      IF(NPRINT .EQ. YES)  WRITE(NUNIT,5002)
C
  440 CALL BKSTEP( BMIV(1,1,NCP1), DA(1,1,NCP1), DV(1,1,NCP1), W, WT,
     1           SYSM, DAMP, STF, CSAVE, NPRINT, KSYS, KPRINT,
     2           DP, Q,QD, QDD,    ALFA, BETA, CONFIG(1,2), F,
     3           ICORD, NCLEAR, AMP, XZERO, DTPM,  DT, ISYS, NC, AMPW,
     4           W2, KSYSM1, KSYSM2, DTW, NSYS, IDT, PIN2$8, IDIM, IQ,
     5           COLM, COLC, COLK, IDP, QSAV, QDSAV, QDDSAV, FSAV,
     6           LAYER, C, IGU, SCALE)
      IF(NPRINT .EQ. YES)  WRITE(NUNIT,5003)
C
C                 ********************************************
C                 **                                        **
C                 **  PROGRAM PARAMETERS ARE NOW CALCULATED  **
C                 **        ESTABLISH PLOT PACKAGE           **
C                 **                                        **
C                 ********************************************
C
  450 CALL FIXFOR(CSAVE, F)
      GO TO 465
C
  463 DO 464  I = 1, KSYS
      FSAV(I) = DABS(F(I))
  464 CONTINUE
  465 IF(DEGR .EQ. 0.0)  CALL DEBUG(2,Q,QD,QDD,F, KSYS, ISYS, DP, NDEG,
     1                                  NPRINT, DEGR, C)
C
      DEGR = WT*DEGOPI
      NDEGKP = NDEG
      NDEG = DEGR
C
      IF( NDEG .EQ. NDEGKP )  GO TO  360
```

```
C
  533 DEG( NDEG )  =  DEGR
      INTIME = NDEG + NTIME
      DEG(INTIME) = DEGR
C
      IF( AGTOUT .EQ. NO ) GO TO  535
C
      IF( NDEG .LT. 360 .OR. NDEG .GT. 725 )   GO  TO  535
      IAGT = IAGT + 1
      IF(IAGT .LT. 2)  GO TO 535
      IAGT = 0
      KAGT = KAGT + 1
C
      CALL AGT(Q, QR4, WT, WTR4, XZERO, ROCKER, NPRINT, AXILTH,
     2             REALTH, C, IUNIT, KAGT)
  535 CONTINUE
C
      CALL ARRAY(F, Q, QD, QDD, NDEG, FPLOT, NTIME)
C
      IF(KPRINT .EQ. NO )  GO TO  634
C
      ICOUNT = ICOUNT + 1
      IF(ICOUNT .LT. NCOUNT)  GO TO  634
      ICOUNT = 0
C
      CALL DEBUG( 2, Q, QD, QDD, F, KSYS, ISYS, DP, NDEG, NPRINT, DEGR,
     1          C)
  634 CONTINUE
C
  636 IF( NDEG .LT. NTIME) GO TO 360
C
C                *******************************************
C                **                                       **
C                **   PLOT PACKAGE FOR SPECIFIED GRAPHS   **
C                **                                       **
C                *******************************************
C
      CALL GRAPH(FPLOT, YNAME, XNAME, DEG1, DEG2, Y1, Y2, Y3, Y4,
     1           YNAME1, YNAME2, PIN2$8, CLEAR,
     2           NTIME, LABEL, TITLE, IDRAW, IPLOT, XSCALE,
     3           IP, NPRINT, GRID, LINES)
C
      GO TO 50
 4037 RETURN
      END
```

SDS-9300 COMPUTER PROGRAM LISTING

This listing is coded in SDS FORTRAN IV, (Ref. 5-7), to run on a Scientific Data System Mode 9300 computer interfaced with an ADAGE Graphics Display Terminal, Model 10 (AGT-10). Due to the limited core storage of this computer, overlaying techniques have been employed.

```
C     SENSE SWITCH 1 - - RESTART - - READ
C     SENSE SWITCH 2 - - INPUT(101)
C     SENSE SWITCH 3 - - CALL CAM
C     SENSE SWITCH 5 - - CALL JOINT
C     SENSE SWITCH 6 - - IF MOD EACH CYCLE
      INTEGER YES, BLNK, PIN28
      INTEGER ROTATE, CONFIG
      DIMENSION IRRAY1( 97),IRRAY2( 18),IRRAY3(13),IRRAY4(32)
      DIMENSION IRRAY5(24), IRRAY6(36),IRRAY7(24),ISPRNG(9),IRRAY9( 49)
      DIMENSION IRRAY10( 49), I101(7), I105(3), I104(8), I117(5)
      DIMENSION I106(12), I107(11), I108( 7), I109( 7), I110(2)
      DIMENSION I111( 5), I112( 9), I113(8), I114(11), I115(1), I116(10)
      DIMENSION X(9), Y(9), CONFIG(180), YZERO(9)
      DIMENSION RADCOS(97), RADSIN(97)
      COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
      COMMON WTH1, WTH2, WTH3, WTH4
      COMMON PI, RAD, DEG, IDEV
      COMMON IGDIR(20), ITDIR(40)
      COMMON XA,YA,XB,YB,XC,YC,XD,YD,XE,YE,XF,YF,XG,YG,XH,YH,XS,YS,XP,YP
      COMMON Q1(180),Q2(180),Q3(180),Q4(180),Q5(180),Q6(180),Q7(180)
      COMMON Q8(180),Q9(180),OMEGA(180)
      COMMON XCAM , YCAM
      DATA NULL/77777777B/, BLNK/4H    /, YES/4HYES /, NO/4HNO  /
  102 FORMAT(1PE12.2,28X)
  103 FORMAT('FREQ. RATIO                                ')
 1430 FORMAT('CONTACT AT :                                    ')
 1431 FORMAT('CAM                                        ')
 1432 FORMAT('ROCKER                                          ')
 1433 FORMAT('NO  CONTACT ')
 1434 FORMAT('VALVE SEAT                   ')
 1450 FORMAT('CAM ROTATION                 ')
 1455 FORMAT('DEGREES ')
 1460 FORMAT('CONFIGURATION                              ')
 1550 FORMAT(I3,1X)
      NAMELIST CONFIG, ISEAT, SF1, SF2, SF3, SF4, SF1A, SF4A, WN, XZERO,
     1  AMP, RPM, IER, RADIUS, DTHETA, FREQ, AXLTH1, AXLTH2, AXLTH3,
     2  AXLTH4, WTH1, WTH2, WTH3, WTH4, IDEV, XA, XB, XC, XD, XE, XF,
     3  XG, XH, XS, XP, XCAM, YA, YB, YC, YD, YE, YF, YG, YH, YS, YP,
     4  YCAM, DIFF
      CMTOGD(A) = A / 0.2000  +  5.000
      GDTOCM(A) = 0.2000*A - 1.0000
      IDEV = 1
    1 CONTINUE
      PI = 3.14159265
      TWOPI = 2.0*PI
      NTIME = 180
      ISEAT = YES
      DEG = 180.0/PI
```

```
      RAD = PI/180.0
      SF1  = 1.0
      SF2  = 1.0
      SF3  = 1.0
      SF4  = 1.0
      SF1A = 15.0
      SF4A = 15.0
      WTH1=WTH2=WTH3=WTH4=0.4
      AXLTH1 = 7.9
      AXLTH2 = 3.4
      AXLTH3 = 5.1
      AXLTH4 = 3.3
      XA = 0.5    ;    YA =-0.5
      XB = 3.9    ;    YB = 9.3
      XC = 10.2   ;    YC = 7.3
      XD = 0.5    ;    YD = 9.1
      XE = 1.0    ;    YE =-0.6
      XF = 0.5    ;    YF = 9.3
      XG = 3.9    ;    YG = 9.1
      XH = 9.0    ;    YH = 9.1
      XP =11.2    ;    YP = -0.02
      XS = 6.8    ;    YS = -0.02
      CALL INITAL(CONFIG, IRRAY1,
     1           IRRAY2,IRRAY3,IRRAY4,IRRAY5,IRRAY6,IRRAY7,IRRAY9,
     2           IRRAY10,ISPRING)
      CALL READ(NTIME, I104, I105, XZERO, AMP, RPM, WN, ISEAT, PIN28,
     1         CONFIG, DIFF)
      THETA = 0.0
      RADIUS = 1.5
      DTHETA= 4.0*RAD
      IF(SENSE SWITCH 2 ) 2, 1260
    2 OUTPUT(101) ' INPUT  CHANGES' ;  INPUT(101)
 1260 CONTINUE
      OUTPUT(6) ' SPRCTR CALLED '
      CALL SPRCTR(IRRAY3,ISEAT, DIFF)
      CALL MIDBAR(IRRAY2,IRRAY9)
      DO  1400  I = 6, 97
      THETA = THETA + DTHETA
      RADCOS(I) = RADIUS*COS(THETA)
      RADSIN(I) = RADIUS*SIN(THETA)
 1400 CONTINUE
      AMPO2 = AMP/2.0
      FREQ = ( RPM * PI ) / ( WN * 30.0 )
      ENCODE(40,102,I116) FREQ
      ENCODE(32,103,I113)
      ENCODE(20,1450,I111)
      ENCODE(8,1455,I110)
      ENCODE(28,1430,I109)
```

```
      ENCODE(32,1431,I104)
      ENCODE(28,1432,I101)
      ENCODE(12,1433,I105)
      ENCODE(20,1434,I117)
      ENCODE(28,1460,I108)
      CALL TEXTO(IDEV,I108, 7,24,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I113,8,38,71,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I116,10,40,65,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I110,2,40,44,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I111,5,38,36,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      N = 0   ;  I = 0
 1500 DO  2000  K = 1, NTIME
      I = I +1
      IF( K .EQ. NTIME)  OUTPUT(6) '           COMPLETE CYCLE'
      N = N +1
      CALL DISPLY(CONFIG, K, I105, I109, I117, I101,I104)
X     OUTPUT(6) ' CAM CALLED '
      IF( SENSE SWITCH 3 ) 3, 1551
    3 CALL CAM( 97,  IRRAY1,RADIUS,AMP,I,RADCOS,RADSIN)
 1551 ROTATE = IFIX(    OMEGA(I) * DEG )
      ENCODE(4,1550,I115) ROTATE
      CALL TEXTO(IDEV,NULL,1,40,36,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I115,1,40,36,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
 1600 CALL BAR1( I,IRRAY4,SF1,SF1A,PIN28)
      CALL BAR2( I,IRRAY5,SF2,IRRAY9,PIN28)
      CALL BAR3( I,IRRAY6,SF3,IRRAY10)
      CALL BAR4( I,IRRAY7,SF4,SF4A)
      IF(  I  .NE. 1 )  GO TO 175
      X(1) = Y(1) = 0.0
      X(2) = XC   ;  Y(2) = YC
      X(3) = X(5) = X(7) = XH - 0.6
      X(4) = X(6) = XH + 0.6
      X(8) = XH
      DO  50  J = 1, 5
      Y(J+2) = FLOAT(J)*(YH-YC)/6.0 + YC
   50 CONTINUE
      Y(8) = YH
      X(9) = Y(9) = 0.0
      ISPRNG(1) = IHEAD(0,10)
      DO  100  J = 1, 9
      YZERO(J) = Y(J)
```

```
      X(J) = GDTOCM( X(J) )  ;  Y(J) = GDTOCM( Y(J) )
  100 CONTINUE
      ISPRNG(2) = IPACK( X(2), Y(2), 0)
      DO   150  J = 3,8
      ISPRNG(J) = IPACK( X(J), Y(J), 1 )
  150 CONTINUE
      ISPRNG(9) = 0
      CALL GRAPHO(IDEV,ISPRNG,9,8,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      IF(  I .EQ. 1 )  GO TO 1990
  175 CONTINUE
      DO 200  J  = 3, 8
      Y(J) = CMTOGD( Y(J) )
      Y(J) = YZERO(J) + FLOAT(J-2) * Q4( I)*SF4/ 6.0
      Y(J) = GDTOCM( Y(J) )
      ISPRNG(J) = IPACK( X(J), Y(J), 1 )
  200 CONTINUE
      CALL GRAPHO(IDEV,ISPRNG,9,8,IER)
      IF(IER.NE.0) OUTPUT(101) IER
 1990 CONTINUE
      IF( SENSE SWITCH 1 ) 1, 1991
 1991 CONTINUE
      IF( SENSE SWITCH 2) 2,1992
 1992 CONTINUE
      IF( SENSE SWITCH 6 ) 6, 2000
    6 CALL TEXTR(IDEV,NULL,1,39,46,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
 1998 IF(MOD(ITDIR(1),8).EQ.0)  GO TO 1998
 2000 CONTINUE
      I = 1  ;  N = 1
      GO TO 1500
      END
```

```
      SUBROUTINE BAR1(JA,LINK1,SF1,SF1A,PIN28)
      INTEGER PIN28, YES
      DIMENSION LINK1(32), X(31), Y(31), AMP(8)
      COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
      COMMON WTH1, WTH2, WTH3, WTH4
      COMMON PI, RAD, DEG, IDEV
      COMMON IGDIR(20), ITDIR(40)
      COMMON XA,YA,XB,YB,XC,YC,XD,YD,XE,YE,XF,YF,XG,YG,XH,YH,XS,YS,XP,YP
      COMMON Q1(180),Q2(180),Q3(180),Q4(180),Q5(180),Q6(180),Q7(180)
      COMMON Q8(180),Q9(180),OMEGA(180)
      COMMON XCAM , YCAM
      DATA YES/4HYES /, NO/4HNO  /
      CMTOGD(A) = A / 0.2000  +  5.000
      GDTOCM(A) = 0.2000*A - 1.0000
      NAMELIST IER
      LINK1(1) = IHEAD(0,10)
      IF( JA .NE. 1 )  GO TO 500
      X(2) = XD   ;   Y(2) = YD
      X(31) = X(2)    ;   Y(31) = Y(2)
      Y(3) = Y(30) = Y(2) - 0.2
      YZERO3 = Y(3)
      DO  30   I = 3, 14
      Y(I) = Y(33-I) = Y(3) - FLOAT(I-3) * AXLTH1 / 11.0
      X(I) = X(2) - WTH1/2.0
      X(33-I) = X(3) + WTH1
   30 CONTINUE
      X(15) = X(16) = X(2) - 3*WTH1
      X(17) = X(18)  =  X(2) + 3*WTH1
      X(30) = X(19)
      Y(15) = Y(18) =Y(19) = Y(14)
      Y(16) = Y(17) = Y(14) - WTH1
      YZER14 = Y(14)
      DO   45  I = 2,31
      X(I) = GDTOCM( X(I) )  ;  Y(I) = GDTOCM( Y(I) )
   45 CONTINUE
   49 DO 50  I = 2,31
      IF( I .EQ. 2)  LINK1(2) = IPACK( X(2), Y(2), 0 )  ;  GO TO 50
      LINK1(I) = IPACK( X(I), Y(I), 1 )
   50 CONTINUE
      LINK1(32) = 0
      CALL GRAPHO( IDEV, LINK1, 32,4,IER )
      IF(IER.NE.0) OUTPUT(101) IER,'BAR1-1'
   99 FORMAT(' ',3X,'I=',I3,2E15.4)
      IF( JA .EQ. 1 )  RETURN
  500 DO  550  I = 2,31
      X(I) = CMTOGD( X(I) )    ;  Y(I) = CMTOGD( Y(I) )
```

```
550 CONTINUE
    IF(PIN28.EQ.YES)  Y(3) = Y(30) = YZERO3 + Q2(JA)*SF1
    IF(PIN28.EQ.NO)   Y(3) = Y(30) = YZERO3 + Q8(JA)*SF1 + YP
    Y(2) = Y(3) + 0.2
    Y(31) = Y(2)
    Y(14) = Y(15) = Y(18) = Y(19) = YZER14  + Q1(JA)*SF1
    Y(16) = Y(17) = Y(14) - WTH1
    DIFF = ABS( Y(3) - Y(14) ) - AXLTH1
    DO  560  I = 4, 13
    Y(I) = Y(33-I)=Y(3)- ( FLOAT(I-3) * (Y(3)-Y(14))/ 11.0 )
560 CONTINUE
    DO  575  I = 4, 8
    AMP(I) = DIFF*SIN( FLOAT(I-3) * PI/11.0 ) * SF1A
    IF( AMP(I) .GT. WTH1/2.0 )     AMP(I) = WTH1/2.0
 98 FORMAT(' ', 3X, 'I= ',I3, '    AMP= ' , E15.5)
    X(I) = X(17-I) = X(3) + AMP(I)
    X(33-I) = X(I+16) = X(I) + WTH1 - 2.0*AMP(I)
575 CONTINUE
    DO  600  I = 2,31
    X(I) = GDTOCM( X(I) )  ;  Y(I) = GDTOCM( Y(I) )
    LINK1(I) = IPACK( X(I), Y(I), 1 )
600 CONTINUE
    LINK1(2) = IPACK( X(2), Y(2), 0 )
    LINK1(32) = 0
    CALL GRAPHO( IDEV, LINK1, 32,4,IER )
    IF(IER.NE.0) OUTPUT(101) IER,'BAR1-2'
    RETURN
    END
```

```
      SUBROUTINE BAR2(JA,LINK2,SF2,IRRAY9,PIN28)
      INTEGER PIN28, YES
      DIMENSION LINK2(24), X(24), Y(24), PHI(10,3), YZERO(24)
      COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
      CCMMON WTH1, WTH2, WTH3, WTH4
      COMMON PI, RAD, DEG, IDEV
      COMMON IGDIR(20), ITDIR(40)
      COMMON XA,YA,XB,YB,XC,YC,XD,YD,XE,YE,XF,YF,XG,YG,XH,YH,XS,YS,XP,YP
      COMMON Q1(180),Q2(180),Q3(180),Q4(180),Q5(180),Q6(180),Q7(180)
      COMMON Q8(180),Q9(180),OMEGA(180)
      COMMON XCAM , YCAM
      DATA YES/4HYES /, NO/4HNO  /
      GDTOCM(A) = 0.2000*A - 1.0000
      CMTOGD(A) = A / 0.2000  +  5.000
      NAMELIST IER
      L = 1
      K = 2
      IF( JA .NE. 1 )  GO TO 200
      X(1) = Y(1) = 0.0
      X(24) = XF    ;  Y(24) = YF
      FACT2 =  Y(24) - WTH2/2.0
      FACT3 =  Y(24) + WTH2/2.0
      DO  10  I = 2,12
      Y(I) = YZERO(I) = FACT2
      Y(I+11) = FACT3
   10 CONTINUE
      DO  25  I = 1, 11
      X(13-I) = X(12+I) = X(24) + FLOAT(I-1) * AXLTH2 / 10.0
   25 CONTINUE
      XA = X(12)          ;   YA = Y(12)
      IF(PIN28 .EQ. NO)  GO TO 26
      IF( SENSE SWITCH 5 ) 5,26
    5 CALL JOINT( 37, IRRAY9, 0.1 ,10.0, K, JA)
   26 CONTINUE
      AL22 = AXLTH2**2
      AL23 = AXLTH2**3
      DO  30  I = 3, 12
      XX = X(I) - X(24)
      XX2 = XX**2
      XX3 = XX2 * XX
      PHI(I-2,1  ) = (AL23 - 3.0*XX2*AXLTH2 + 2.0*XX3 ) * SF2/AL23
      PHI(I-2,3-L) = (XX*AL22 - 2.0*XX2*AXLTH2 + XX3 ) * SF2/AL22
      PHI(I-2,4-L) = -( XX2*AXLTH2 - XX3 ) * SF2/AL22
   30 CONTINUE
      DO  50  I = 1, 23
   99 FORMAT(' ',3X,'I=',I3,2E15.4)
```

```
      X(I) = GDTOCM( X(I) )  ;  Y(I) = GDTOCM( Y(I) )
   50 CONTINUE
      LINK2(1) = IHEAD(0,10)
      LINK2(2) = IPACK( X(2), Y(2), 0 )
      DO  100  I = 3, 23
      LINK2(I) = IPACK( X(I), Y(I), 1 )
  100 CONTINUE
      LINK2(24) = 0
      CALL GRAPHO(IDEV, LINK2,24,5,IER)
      IF(IER.NE.0) OUTPUT(101) IER,'BAR2-1'
      IF( JA .EQ. 1 )  RETURN
  200 CONTINUE
      DO  300  I = 3, 12
      Y(I) = CMTOGD( Y(I) )
      Y(25-I) = CMTOGD( Y(25-I) )
      W = PHI(I-2,1)*Q2(JA) + PHI(I-2,3-L)*Q3(JA) + PHI(I-2,4-L)*Q6(JA)
      Y(I) = W + YZERO(I)
      Y(25-I) = WTH2 + Y(I)
  300 CONTINUE
      X(12) = CMTOGD( X(12) )
      XA = X(12)         ;    YA = Y(12)
      IF(PIN28 .EQ. NO)  GO TO 326
      IF( SENSE SWITCH 5 )  325,326
  325 CALL JOINT( 37, IRRAY9, 0.1 ,10.0, K, JA)
  326 CONTINUE
      X(12) = GDTOCM( X(12) )
      DO  350  I  = 3,12
      Y(I) = GDTOCM( Y(I) )
      Y(25-I) = GDTOCM( Y(25-I) )
  350 CONTINUE
      DO  400  I = 3, 23
      LINK2(I) = IPACK( X(I), Y(I), 1 )
  400 CONTINUE
      CALL GRAPHO(IDEV, LINK2,24,5,IER)
      IF(IER.NE.0) OUTPUT(101) IER,'BAR2-2'
      RETURN
      END
```

```
      SUBROUTINE BAR3(JA,LINK3,SF3,IRRAY10)
      DIMENSION LINK3(36), X(35), Y(35), PHI(16,3), YZERO(35)
      CCMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
      COMMON WTH1, WTH2, WTH3, WTH4
      COMMON PI, RAD, DEG, IDEV
      CCMMON IGDIR(20), ITDIR(40)
      COMMON XA,YA,XB,YB,XC,YC,XD,YD,XE,YE,XF,YF,XG,YG,XH,YH,XS,YS,XP,YP
      COMMON Q1(180),Q2(180),Q3(180),Q4(180),Q5(180),Q6(180),Q7(180)
      COMMON Q8(180),Q9(180),OMEGA(180)
      CCMMON XCAM , YCAM
      GDTOCM(A) = 0.2000*A - 1.0000
      CMTOGD(A) = A / 0.2000  +  5.000
      NAMELIST IER
      L = 1
      K = 3
      IF( JA .NE. 1 )  GO TO 200
      X(1) = Y(1) = 0.0
      X(2) = XG   ;   Y(2) = YG
      DO  10  I = 2,18
      Y(I) = YZERO(I) = Y(2)
      Y(I+17) = Y(2) + WTH3
   10 CONTINUE
      DO  25  I = 2, 18
      X(I) = X(37-I) = X(2) + FLOAT(I-2) * AXLTH3 / 16.0
   25 CONTINUE
      XA = X(18)  ;  YA = Y(18)
      IF( SENSE SWITCH 5 ) 5,26
    5 CALL JOINT( 37, IRRAY10,0.1 ,10.0, K, JA)
   26 CONTINUE
      AL32 = AXLTH3**2
      AL33 = AXLTH3**3
      DO  30  I = 3, 18
      XX = X(I) - X(2)
      XX2 = XX**2
      XX3 = XX2 * XX
      PHI(I-2,2-L) =( + 3.0*XX2*AXLTH3 - 2.0*XX3 ) * SF3/AL33
      PHI(I-2,3-L) = ( XX*AL32 - 2.0*XX2*AXLTH3 + XX3 ) * SF3/AL32
      PHI(I-2,4-L) = -( XX2*AXLTH3 - XX3 ) * SF3/AL32
   30 CONTINUE
      DO  50  I = 1, 35
      X(I) = GDTOCM( X(I) )  ;  Y(I) = GDTOCM( Y(I) )
   50 CONTINUE
      LINK3(1) = IHEAD(0,10)
      LINK3(2) = IPACK( X(2), Y(2), 0 )
      DO  100  I = 3, 35
      LINK3(I) = IPACK( X(I), Y(I), 1 )
```

```
100 CONTINUE
    LINK3(36) = 0
    CALL GRAPHO(IDEV, LINK3,36, 6, IER )
    IF(IER.NE.0) OUTPUT(101) IER,'BAR3-1'
    IF( JA .EQ. 1 )  RETURN
200 CONTINUE
    DO  300  I = 3, 18
    Y(I) = CMTOGD( Y(I) )
    Y(37-I) = CMTOGD( Y(37-I) )
    W= PHI(I-2,2-L)*Q4(JA) + PHI(I-2,3-L)*Q6(JA) + PHI(I-2,4-L)*Q5(JA)
    Y(I) = W + YZERO(I)
    Y(37-I) = WTH3 + Y(I)
300 CONTINUE
    X(18) = CMTOGD( X(18) )
    XA = X(18)  ;  YA = Y(18)
    IF( SENSE SWITCH 5 )  325,326
325 CALL JOINT( 37, IRRAY10,0.1 ,10.0, K, JA)
326 CONTINUE
    X(18) = GDTOCM( X(18) )
    DO  350  I = 3, 18
    Y(I) = GDTOCM( Y(I) )
    Y(37-I) = GDTOCM( Y(37-I) )
350 CONTINUE
    DO  400  I = 3, 35
    LINK3(I) = IPACK( X(I), Y(I), 1 )
400 CONTINUE
    CALL GRAPHO(IDEV, LINK3,36,6,IER)
    IF(IER.NE.0) OUTPUT(101) IER,'BAR3-2'
    RETURN
    END
```

```
      SUBROUTINE BAR4(JA,LINK4,SF4,SF4A)
      DIMENSION LINK4(24), X(23), Y(23), AMP(7)
      COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
      COMMON WTH1, WTH2, WTH3, WTH4
      COMMON PI, RAD, DEG, IDEV
      COMMON IGDIR(20), ITDIR(40)
      COMMON XA,YA,XB,YB,XC,YC,XD,YD,XE,YE,XF,YF,XG,YG,XH,YH,XS,YS,XP,YP
      COMMON Q1(180),Q2(180),Q3(180),Q4(180),Q5(180),Q6(180),Q7(180)
      COMMON Q8(180),Q9(180),OMEGA(180)
      COMMON XCAM , YCAM
      GDTOCM(A) = 0.2000*A - 1.0000
      CMTOGD(A) = A / 0.2000  +  5.000
      NAMELIST IER
      LINK4(1) = IHEAD(0,10)
      IF( JA .NE. 1 )  GO TO 500
      X(1) = Y(1) = 0.0
      X(2) = XH    ;    Y(2) = YH
      X(23) = X(2)    ;   Y(23) = Y(2)
      X(3) = X(2) - WTH4/2.0
      Y(3) = Y(2) - WTH4/2.0
      YZERO3 = Y(3)
      DO  25   I = 3, 11
      X(I) = X(3)
      X(25-I) = X(3) + WTH4
      Y(I) = Y(25-I) = Y(3) - AXLTH4 * FLOAT(I-3) / 8.0
   25 CONTINUE
      X(12) = X(11) - 1.2
      X(13) = X(14) + 1.2
      Y(12) = Y(13) = Y(11) - 0.3
      YZER11 = Y(11)
      DO  100  I = 2, 23
      X(I) = GDTOCM( X(I) )  ;  Y(I) = GDTOCM( Y(I) )
  100 CONTINUE
      DO  150  I = 2, 23
      IF( I .EQ. 2)  LINK4(2) = IPACK( X(2), Y(2), 0 )  ;  GO TO 150
      LINK4(I) = IPACK( X(I), Y(I), 1 )
  150 CONTINUE
      LINK4(24) = 0
      CALL GRAPHO(IDEV,LINK4,24,7,IER)
      IF(IER.NE.0) OUTPUT(101) IER,'BAR4-1'
      DO  275  I = 1,22
  275 CONTINUE
      IF( JA .EQ. 1 )  RETURN
  500 DO  550  I = 2,23
      X(I) = CMTOGD( X(I) )    ;  Y(I) = CMTOGD( Y(I) )
  550 CONTINUE
```

```
      Y(3) = Y(22) =  YZERO3 + Q4(JA)*SF4
      Y(2) = Y(3) + 0.2
      Y(23) = Y(2)
      Y(11) = Y(14) = YZER11 -  Q7(JA)*SF4
      Y(12) = Y(13) = Y(11) - 0.3
      DIFF = ABS( Y(3) - Y(11) ) - AXLTH4
      DO  560  I = 4, 10
      Y(I)  = Y(25-I) = Y(3) - ( FLOAT(I-3) * (Y(3)-Y(11)) / 8.0)
  560 CONTINUE
      DO  575  I = 4, 7
      AMP(I) = DIFF*SIN( FLOAT(I-3) * PI/8.0 ) * SF4A
      IF( AMP(I)  .GT. WTH4/2.0 )    AMP(I) = WTH4/2.0
      X(I) = X(14-I) =  X(3) + AMP(I)
      X(25-I) = X(I+11) = X(I) + WTH4 - 2.0 * AMP(I)
  575 CONTINUE
      DO  600  I = 2,23
      X(I) = GDTOCM( X(I) )  ;  Y(I) = GDTOCM( Y(I) )
      LINK4(I) = IPACK( X(I), Y(I), 1 )
  600 CONTINUE
      LINK4(2) = IPACK( X(2), Y(2), 0 )
      LINK4(24) = 0
      CALL GRAPHO(IDEV,LINK4,24,7,IER)
      IF(IER.NE.0) OUTPUT(101) IER,'BAR4-2'
      RETURN
      END
```

```
      SUBROUTINE CAM(NCAM,ICAM,RADIUS,AMP,JA,RADCOS,RADSIN)
      DIMENSION ICAM(NCAM), X(97), Y(97), RADCOS(97), RADSIN(97)
      COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
      COMMON WTH1, WTH2, WTH3, WTH4
      COMMON PI, RAD, DEG, IDEV
      COMMON IGDIR(20), ITDIR(40)
      COMMON XA,YA,XB,YB,XC,YC,XD,YD,XE,YE,XF,YF,XG,YG,XH,YH,XS,YS,XP,YP
      COMMON Q1(180),Q2(180),Q3(180),Q4(180),Q5(180),Q6(180),Q7(180)
      COMMON Q8(180),Q9(180),OMEGA(180)
      COMMON XCAM , YCAM
      CMTOGD(A) = A / 0.2000  +  5.000
      GDTOCM(A) = 0.2000*A - 1.0000
      X(1) = 0.0         ;    Y(1) = 0.0
      IF( JA .NE. 1 )  GO TO 55
      ICAM(1) = IHEAD(0,10)
      CROSS = 0.3
      XCAM = XE - AMP
      X(2) = XCAM + CROSS
      X(3) = XCAM - CROSS
      YCAM = YD - 0.6 - AXLTH1 - RADIUS + YS
      Y(2) = YCAM + CROSS ;  Y(3) = YCAM - CROSS
      X(4) = X(5) = Y(4) = Y(5) = 0.0
      DO  30  I = 1,3
      X(I) = GDTOCM( X(I) )  ;  Y(I) = GDTOCM( Y(I) )
   30 CONTINUE
      XCAM = GDTOCM( XCAM )  ;  YCAM = GDTOCM( YCAM )
      ICAM(2) = IPACK( X(2), YCAM, 0 )
      ICAM(3) = IPACK( X(3), YCAM, 1 )
      ICAM(4) = IPACK( XCAM, Y(2), 0 )
      ICAM(5) = IPACK( XCAM, Y(3), 1 )
   55 CONTINUE
      IF( JA .EQ. 1 )  XCAM = CMTOGD( XCAM )    ;  YCAM = CMTOGD( YCAM )
      XE = XCAM + COS(OMEGA(JA))
      YE = Q9(JA) + YCAM
      X(6) = RADIUS + XE  ;  Y(6) = YE
      DO 200 I = 6, NCAM-1
      X(I+1) = RADCOS(I+1) + XE
      Y(I+1) = RADSIN(I+1) + YE
      X(I) = GDTOCM( X(I) )  ;  Y(I) = GDTOCM( Y(I) )
      IF(I.EQ.6) ICAM(6)= IPACK(X(6),Y(6),0); GO TO 200
      ICAM(I)= IPACK(X(I),Y(I),1)
  200 CONTINUE
      ICAM(NCAM) = 0
      CALL GRAPHO(IDEV,ICAM,NCAM,1,IER)
      RETURN
      END
```

```
      SUBROUTINE  DISPLY(CONFIG, K, I105, I109, I117, I101,I104)
      DIMENSION I101(7), I104(8), I105(3), I109(7), I117(5), CONFIG(180)
      INTEGER ROTATE, CONFIG, YES, BLNK
      COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
      COMMON WTH1, WTH2, WTH3, WTH4
      COMMON PI, RAD, DEG, IDEV
      COMMON IGDIR(20), ITDIR(40)
      COMMON XA,YA,XB,YB,XC,YC,XD,YD,XE,YE,XF,YF,XG,YG,XH,YH,XS,YS,XP,YP
      COMMON Q1(180),Q2(180),Q3(180),Q4(180),Q5(180),Q6(180),Q7(180)
      COMMON Q8(180),Q9(180),OMEGA(180)
      COMMON XCAM , YCAM
      DATA NULL/77777777B/, BLNK/4H    /, YES/4HYES /, NO/4HNO  /
      NAMELIST IER
      GO TO(1501,1502,1503,1504,1505,1506,1507,1508) CONFIG(K)
 1501 CONTINUE
      CALL TEXTO(IDEV,NULL, 1,26,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I105, 3,26,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,NULL, 1,28,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,NULL, 1,30,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,NULL, 1,32,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      GO TO 1509
 1502 CONTINUE
      CALL TEXTO(IDEV,NULL, 1,26,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I109, 3,26,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,NULL, 1,28,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,NULL, 1,30,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I117, 5,32,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      GO TO 1509
 1503 CONTINUE
      CALL TEXTO(IDEV,NULL, 1,26,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I109, 3,26,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I101, 7,28,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,NULL, 1,30,30,2,3,IER)
```

```
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,NULL, 1,32,30,2,3,IER)
      GO TO 1509
 1504 CONTINUE
      CALL TEXTO(IDEV,NULL, 1,26,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I109, 3,26,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I101, 7,28,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,NULL, 1,30,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I117, 5,32,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      GO TO 1509
 1505 CONTINUE
      CALL TEXTO(IDEV,NULL, 1,26,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I109, 3,26,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,NULL, 1,28,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I104, 8,30,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,NULL, 1,32,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      GO TO 1509
 1506 CONTINUE
      CALL TEXTO(IDEV,NULL, 1,26,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I109, 3,26,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,NULL, 1,28,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I104, 8,30,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I117, 5,32,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      GO TO 1509
 1507 CONTINUE
      CALL TEXTO(IDEV,NULL, 1,26,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I109, 3,26,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I101, 7,28,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I104, 8,30,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
```

```
      CALL TEXTO(IDEV,NULL, 1,32,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      GO TO 1509
 1508 CONTINUE
      CALL TEXTO(IDEV,NULL, 1,26,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I109, 3,26,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I101, 7,28,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I104, 8,30,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I117, 5,32,30,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
 1509 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INITAL(CONFIG, IRRAY1,
     1           IRRAY2,IRRAY3,IRRAY4,IRRAY5,IRRAY6,IRRAY7,IRRAY9,
     2           IRRAY10,ISPRING)
      INTEGER CONFIG, YES, BLNK
      DIMENSION IRRAY1( 97),IRRAY2( 18),IRRAY3(13),IRRAY4(32)
      DIMENSION IRRAY5(24), IRRAY6(36),IRRAY7(24),ISPRNG(9),IRRAY9( 49)
      DIMENSION IRRAY10( 49), I101(7)
      DIMENSION I106(12), I107(11), I108( 7), I109( 7), I110(2)
      DIMENSION I111( 5), I112( 9), I113( 8), I114(18), I115(1),I116(10)
      DIMENSION CONFIG(180), I117(5)
      COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
      COMMON WTH1, WTH2, WTH3, WTH4
      COMMON PI, RAD, DEG, IDEV
      COMMON IGDIR(20), ITDIR(40)
      COMMON XA,YA,XB,YB,XC,YC,XD,YD,XE,YE,XF,YF,XG,YG,XH,YH,XS,YS,XP,YP
      COMMON Q1(180),Q2(180),Q3(180),Q4(180),Q5(180),Q6(180),Q7(180)
      COMMON Q8(180),Q9(180),OMEGA(180)
      COMMON XCAM , YCAM
      DATA NULL/77777777B/, BLNK/4H    /, YES/4HYES /, NO/4HNO  /
      NAMELIST IER
  101 FORMAT('  CAM ACTUATED VALVE TRAIN  ')
  106 FORMAT('  THIS PROGRAM SIMULATES AN ELASTIC LINK MODEL  ')
  107 FORMAT(' OF AN AUTOMOBILE CAM ACTUATED VALVE TRAIN ')
  108 FORMAT(' LT. CHARLES W. GNILKA USN  ')
  109 FORMAT(' VILLANOVA UNIVERSITY 1964  ')
  110 FORMAT('ADVISOR ')
  111 FORMAT('DAVID  SALINAS, PHD ')
  112 FORMAT('PROFESSOR OF MECHANICAL ENGINEERING ')
  113 FORMAT('U. S. NAVAL POSTGRADUATE SCHOOL ')
  114 FORMAT('FORMER  ADVISOR:  RICHARD  C.  WINFREY, PHD ')
  115 FORMAT(' BY ')
  116 FORMAT('NAVAL CIVIL ENG. LAB, PORT HUENEME, CAL.')
  117 FORMAT(' WITH  CLEARANCES   ')
      IF(IDEV.EQ.1) OUTPUT(6) '   AGT-1'
      IF(IDEV.EQ.2) OUTPUT(6) '   AGT-2'
      OUTPUT(101) IDEV,'INPUT CHANGES' ;   INPUT(101)
      DO 10 JA =1,20
   10 IGDIR(JA) = 0
      CALL DGINIT(IDEV,IGDIR,20,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL DTINIT(IDEV,ITDIR,40,IER)
      IF(IER.NE.0) OUTPUT(101) IER
C     INITIALIZE TEXTI BLOCK
      CALL TEXTO(IDEV,NULL,1,39,46,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      ENCODE(28,101,I101)
```

```
      ENCODE(48,106,I106)
      ENCODE(44,107,I107)
      ENCODE(28,108,I108)
      ENCODE(28,109,I109)
      ENCODE( 8,110,I110)
      ENCODE(20,111,I111)
      ENCODE(36,112,I112)
      ENCODE(32,113,I113)
      ENCODE(44,114,I114)
      ENCODE( 4,115,I115)
      ENCODE(40,116,I116)
      ENCODE(20, 117,I117)
      CALL TEXTO(IDEV,I101,7,1,13,3,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, I106,12, 5, 1,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, I107,11, 8, 6,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I115,1,14,45,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, I108, 7,17,25,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, I109, 7,19,37,1,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, I110, 2,24,43,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, I111, 5,27,29,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, I112, 9,29,31,1,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, I113, 8,30,33,1,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, I114,11,33,30,1,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, I116,10,34,33,1,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, I117, 5,11,27,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
C     PAUSE TO READ TITLE PAGE
      CALL TEXTR(IDEV,NULL,1,39,46,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
   12 IF(MOD(ITDIR(1),8).EQ.0)  GO TO 12
      CALL TEXTO(IDEV,NULL,1,39,46,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      DO 20 JA = 1,97
   20 IRRAY1(JA) = 0.0
      DO 21 JA = 1,18
   21 IRRAY2(JA) = 0.0
```

```
      DO 22 JA = 1,11
   22 IRRAY3(JA) = 0.0
      DO 23 JA = 1,32
   23 IRRAY4(JA) = 0.0
      DO  24  JA = 1,24
   24 IRRAY5(JA) = 0.0
      DO  25  JA = 1,36
   25 IRRAY6(JA) = 0.0
      DO  26  JA = 1,24
   26 IRRAY7(JA) = 0.0
      DO  27  JA = 1,9
   27 ISPRNG(JA) = 0.0
      DO  28  JA = 1,49
   28 IRRAY9(JA) = 0.0
      DO  29  JA = 1,49
   29 IRRAY10(JA) = 0.0
      IRRAY1(1) = IHEAD(0,10)
      IRRAY2(1) = IHEAD(0,10)
      IRRAY3(1) = IHEAD(0,10)
      IRRAY4(1) = IHEAD(0,10)
      IRRAY5(1) = IHEAD(0,10)
      IRRAY6(1) = IHEAD(0,10)
      IRRAY7(1) = IHEAD(0,10)
      ISPRNG(1) = IHEAD(0,10)
      IRRAY9(1) = IHEAD(0,10)
      IRRAY10(1) = IHEAD(0,10)
      CALL GRAPHO (IDEV,IRRAY1, 97,1,IER)
      IF(IER.NE.0) OUTPUT(101) IER,'IRRAY1'
      CALL GRAPHO (IDEV,IRRAY2, 18,2,IER)
      IF(IER.NE.0) OUTPUT(101) IER,'IRRAY2'
      CALL GRAPHO (IDEV,IRRAY3, 13,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER,'IRRAY3'
      CALL GRAPHO (IDEV,IRRAY4, 32,4,IER)
      IF(IER.NE.0) OUTPUT(101) IER,'IRRAY4'
      CALL GRAPHO (IDEV,IRRAY5, 24,5,IER)
      IF(IER.NE.0) OUTPUT(101) IER,'IRRAY5'
      CALL GRAPHO (IDEV,IRRAY6, 36,6,IER)
      IF(IER.NE.0) OUTPUT(101) IER,'IRRAY6'
      CALL GRAPHO(IDEV,IRRAY7,24,7,IER)
      IF(IER.NE.0) OUTPUT(101) IER,'IRRAY7'
      CALL GRAPHO(IDEV,ISPRNG,9,8,IER)
      IF(IER.NE.0) OUTPUT(101) IER,'IRRAY8'
      CALL GRAPHO (IDEV,IRRAY9, 49,9,IER)
      IF(IER.NE.0) OUTPUT(101) IER,'IRRAY9'
      CALL GRAPHO (IDEV,IRRAY10, 49,10,IER)
      IF(IER.NE.0) OUTPUT(101) IER,'IRRAY10'
      RETURN
      END
```

```
      SUBROUTINE JOINT(NCIR,ICIR, RADIUS, ANGLE, K, JA)
      DIMENSION ICIR(NCIR), X( 75), Y( 75), RADCOS( 75), RADSIN( 75)
      COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
      COMMON WTH1, WTH2, WTH3, WTH4
      COMMON PI, RAD, DEG, IDEV
      COMMON IGDIR(20), ITDIR(40)
      COMMON XA,YA,XB,YB,XC,YC,XD,YD,XE,YE,XF,YF,XG,YG,XH,YH,XS,YS,XP,YP
      COMMON Q1(180),Q2(180),Q3(180),Q4(180),Q5(180),Q6(180),Q7(180)
      COMMON Q8(180),Q9(180),OMEGA(180)
      COMMON XCAM , YCAM
      GDTOCM(A) = 0.2000*A - 1.0000
      NAMELIST IER
      IF( JA .NE. 1 )  GO TO 35
      ICIR(1) = IHEAD(1,10)
      X(1)=Y(1)=0.0
      THETA = 0.0
      DTHETA= ANGLE * RAD
      DO  25  I = 3, NCIR
      THETA = THETA + DTHETA
      RADCOS(I) = RADIUS*COS(THETA)
      RADSIN(I) = RADIUS*SIN(THETA)
   25 CONTINUE
   35 CONTINUE
      X(2) = RADIUS + XA  ;  Y(2) = YA
      DO 100 I = 3,NCIR
      X(I) = RADCOS(I) + XA
      Y(I) = RADSIN(I) + YA
  100 CONTINUE
X     OUTPUT(6) '    JOINT CALLED'
      DO 200 I = 2, NCIR
X     WRITE(6,99) I, X(I), Y(I)
X  99 FORMAT(' ',3X,'I=',I3,2E15.4)
      X(I) = GDTOCM( X(I) )  ;  Y(I) = GDTOCM( Y(I) )
      IF(I.EQ.2) ICIR(2)= IPACK(X(2),Y(2),0); GO TO 200
      ICIR(I)= IPACK(X(I),Y(I),1)
 200  CONTINUE
      CALL GRAPHO(IDEV,ICIR,NCIR,7+K,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      RETURN
      END
```

```
      SUBROUTINE MIDBAR(MIDCIR,IRRAY9)
      DIMENSION MIDCIR(18), X(18), Y(18), IRRAY9(49)
      COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
      COMMON WTH1, WTH2, WTH3, WTH4
      COMMON PI, RAD, DEG, IDEV
      COMMON IGDIR(20), ITDIR(40)
      COMMON XA,YA,XB,YB,XC,YC,XD,YD,XE,YE,XF,YF,XG,YG,XH,YH,XS,YS,XP,YP
      COMMON Q1(180),Q2(180),Q3(180),Q4(180),Q5(180),Q6(180),Q7(180)
      COMMON Q8(180),Q9(180),OMEGA(180)
      COMMON XCAM , YCAM
      NAMELIST IER
      GDTOCM(A) = 0.2000*A - 1.0000
      MIDCIR(1) = IHEAD(0,10)
      X(1)=Y(1)=0.0
      ANGLE = 9.0  ;  K = 6   ;  JA = 1
      RADIUS= 0.25
      XA = XB   ;   YA = YB
      CALL JOINT(49, IRRAY9, RADIUS, ANGLE,K, JA)
      DO  100  I = 1, 49
      IRRAY9(I) = 0
  100 CONTINUE
      X(  2) = XB ; Y(  2) = YB
      X(  3) = XB + .5  ;  Y(  3) = YB - .5
      X(  4) = X(  3) - 1.0  ;  Y(  4) = Y(  3)
      X(  5) = X(  2)  ;  Y(  5) = Y(  2)
      X(  6) = X(  3)  ;  Y(  6) = Y(  3)
      DO 150 I = 1,11
      X(  6 + I) = X(  5+I) - .1
  150 CONTINUE
      DO  160  I = 7,17,2
      Y(I) = Y(  6) - .1
      Y(I+1) = Y(  6)
  160 CONTINUE
      DO 200 I = 2, 17
      X(I) = GDTOCM( X(I) )  ;  Y(I) = GDTOCM( Y(I) )
      IF(I.EQ.2 )  MIDCIR(I)=IPACK(X(I),Y(I),0); GO TO 200
      MIDCIR(I) = IPACK(X(I),Y(I),1)
  200 CONTINUE
      DO 250 I = 8, 18, 2
      MIDCIR(I) = IPACK(X(I),Y(I),0)
  250 CONTINUE
      MIDCIR( 18) = 0
      CALL GRAPHO(IDEV,MIDCIR, 18,2,IER)
X     IF(IER.NE.0) OUTPUT(101) IER
      RETURN
      END
```

```
      SUBROUTINE READ(NTIME, I104, I105, XZERO, AMP, RPM, WN, ISEAT,
     1                PIN28, CONFIG, DIFF)
      INTEGER CONFIG, YES, BLNK
      DIMENSION I104(8), I105(3), CONFIG(180)
      COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
      COMMON WTH1, WTH2, WTH3, WTH4
      COMMON PI, RAD, DEG, IDEV
      COMMON IGDIR(20), ITDIR(40)
      COMMON XA,YA,XB,YB,XC,YC,XD,YD,XE,YE,XF,YF,XG,YG,XH,YH,XS,YS,XP,YP
      COMMON Q1(180),Q2(180),Q3(180),Q4(180),Q5(180),Q6(180),Q7(180)
      COMMON Q8(180),Q9(180),OMEGA(180)
      COMMON XCAM , YCAM
      DATA NULL/77777777B/, BLNK/4H    /, YES/4HYES /, NO/4HNO  /
      NAMELIST IER
   30 FORMAT(5E15.7)
   31 FORMAT(5E15.7, I3)
   40 FORMAT(' ', 4X, I3, 2X, 9(2X,1PE10.3), 6X, I2)
   45 FORMAT('0', 2X, 'DEGREES', 6X, 'Q1', 10X, 'Q2', 10X, 'Q3', 10X,
     1       'Q4', 10X, 'Q5', 10X, 'Q6', 10X, 'Q7', 10X, 'Q8', 8X,
     2       'OMEGA', 7X, 'CONFIG', //)
  104 FORMAT(' LISTING OF COORDINATES DESIRED ')
  105 FORMAT('  COMMAND:  ')
 5012 FORMAT('1', 20X, 20A4)
 5013 FORMAT('0', 20X, 20A4)
 5014 FORMAT(18A4)
 5015 FORMAT(5E10.4, 2A4)
 5016 FORMAT('0',  ' XZERO =',E10.4,'      AMP =',E10.4,'    RPM=',E10.4,
     1 '      NAT. FREQ =',3PE10.1,6X,1A4,' VALVE SEAT ', 1A4,
     2   ' COORD. 2 AND 8 PINNED')
 5017 FORMAT('0', 10X, 'AT ZERO DEGREES - - CLEARANCE BETWEEN VALVE'
     1       ' AND VALVE SEAT IS ', F7.3)
    4 REWIND 4
      READ(4,5014) (Q1(JA),JA=1,18)
      READ(4,5014) (Q2(JA),JA=1,18)
      READ(4,5015) XZERO,AMP,RPM,WN,CLEAR, ISEAT, PIN28
      WRITE(6,5012) (Q1(JA),JA=1,18)
      WRITE(6,5013) (Q2(JA),JA=1,18)
      WRITE(6,5016) XZERO,AMP,RPM,WN,ISEAT,PIN28
C     DESIRE COORDINATES PRINTED
      ENCODE(32,104,I104)
      ENCODE(12,105,I105)
      CALL TEXTO(IDEV,I104,8,37,17,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,I105,3,39,23,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTR(IDEV,NULL,1,39,46,2,3,IER)
```

```
      IF(IER.NE.0) OUTPUT(101) IER
   15 IF(MOD(ITDIR(1),8).EQ.0)  GO TO 15
      CALL TEXTI(IDEV,IMODA,1,39,46,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      IPRINT = NO
      IF( IMODA .EQ. YES )  IPRINT = YES
C     ERASE QUESTION
      CALL TEXTO(IDEV,NULL,1,39,46,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,NULL,1,39,23,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,NULL,1,37,17,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      READ(4,30) DIFF
      WRITE(6,5017) DIFF
      DO  1000  I = 1,NTIME
      READ(4,30) Q1(I),Q2(I),Q3(I),Q4(I),Q5(I)
      READ(4,31) Q6(I),Q7(I),Q8(I),Q9(I),OMEGA(I),CONFIG(I)
 1000 CONTINUE
      IF( IPRINT .EQ. NO )   GO TO 1250
 1100 WRITE(6,45)
      DO  1200  I = 1, NTIME
      NDEG = IFIX( OMEGA(I)*DEG )
      WRITE(6,40) NDEG,Q1(I),Q2(I),Q3(I),Q4(I),Q5(I),Q6(I),Q7(I),
     1            Q8(I),OMEGA(I), CONFIG(I)
 1200 CONTINUE
 1250 CONTINUE
C     NULL TITLE PAGE
      CALL TEXTO(IDEV, NULL, 1, 5, 1,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, NULL, 1, 8, 6,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV,NULL,1,14,45,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, NULL, 1,17,25,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, NULL, 1,19,37,1,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, NULL, 1,24,43,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, NULL, 1,27,29,2,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, NULL, 1,29,31,1,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, NULL, 1,30,33,1,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTO(IDEV, NULL, 1,33,30,1,3,IER)
      IF(IER.NE.0) OUTPUT(101) IER
```

```
CALL TEXTO(IDEV, NULL, 1,34,33,1,3,IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL TEXTO(IDEV, NULL, 1,11,27,2,3,IER)
IF(IER.NE.0) OUTPUT(101) IER
RETURN
END
```

```
      SUBROUTINE SPRCTR(ISPRNG,ISEAT, DIFF)
      DIMENSION X(13), Y(13),ISPRNG(13)
      INTEGER YES
      COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
      COMMON WTH1, WTH2, WTH3, WTH4
      COMMON PI, RAD, DEG, IDEV
      COMMON IGDIR(20), ITDIR(40)
      COMMON XA,YA,XB,YB,XC,YC,XD,YD,XE,YE,XF,YF,XG,YG,XH,YH,XS,YS,XP,YP
      COMMON Q1(180),Q2(180),Q3(180),Q4(180),Q5(180),Q6(180),Q7(180)
      COMMON Q8(180),Q9(180),OMEGA(180)
      COMMON XCAM , YCAM
      DATA YES/4HYES /, NO/4HNO  /
      NAMELIST IER
      GDTOCM(A) = 0.2000*A - 1.0000
      ISPRNG(1)  = IHEAD(0,10)
      X(1)=Y(1) =0.0
      X(2) = XC  ; Y(2) = YC -.3
      X(3) = XC  ; Y(3) = YC +.3
      DO 100 I = 1, 7
      Y(3+I) = Y(2+I) - .1
      X(I+3) = X(I+2) - .1*(-1.0)**(I+2)
  100 CONTINUE
      DO 175 I = 1,10
      X(I) = GDTOCM( X(I) )  ;  Y(I) = GDTOCM( Y(I) )
  175 CONTINUE
      DO  200 I = 2,10
      IF(I .EQ. 2) ISPRNG(I) = IPACK( X(I), Y(I), 0);  GO TO 200
      ISPRNG(I) = IPACK( X(I), Y(I), 1 )
  200 CONTINUE
      DO 250 I = 5,9,2
      ISPRNG(I) = IPACK( X(I), Y(I), 0 )
  250 CONTINUE
      ISPRNG(11) = 0
      CALL GRAPHO(IDEV, ISPRNG, 11,3, IER)
      IF(IER.NE.0) OUTPUT(101) IER,'SPRCTR'
      IF( ISEAT .EQ. NO)  DIFF = 1.2
      K = 0
      SIGN = +1.0
      X(12) = XS
      Y(2) = YH - WTH4/2.0 - AXLTH4 + DIFF + YP
      Y(3) = Y(2) - 0.4
      Y(4) = Y(2)
      DO  275  I = 4,10,2
      Y(I+1) = Y(I+2) = Y(I) - .1
  275 CONTINUE
  300 K = K + 1
```

```
      DO  350  I = 1,7,2
      X(12-I) = X(13-I) + SIGN*0.3
      X(11-I) = X(12-I) + SIGN*0.1
  350 CONTINUE
      X(3) = X(11)
      X(2) = X(4) + 0.3*SIGN
      X(13) = Y(13) = 0.5
      DO  375  I = 1,13
      X(I) = GDTOCM( X(I) )
      IF( K .EQ. 1 )  Y(I) = GDTOCM( Y(I) )
  375 CONTINUE
      ISPRNG(2) = IPACK( X(2), Y(2), 0 )
      ISPRNG(3) = IPACK( X(3), Y(3), 1 )
      ISPRNG(4) = IPACK( X(4), Y(4), 0 )
      ISPRNG(5) = IPACK( X(2), Y(2), 1 )
      DO  400  I = 6, 12, 2
      ISPRNG(I) = IPACK( X(I-1), Y(I-1), 0 )
      ISPRNG(I+1) = IPACK( X(I), Y(I), 1 )
  400 CONTINUE
      ISPRNG(13) = 0
      CALL GRAPHO(IDEV, ISPRNG, 13,10+K, IER)
X     IF(IER.NE.0) OUTPUT(101) IER
      X(12) = XP
      SIGN = -1.0
      IF( K .EQ. 1 ) GO TO 300
      RETURN
  500 DO  600  I = 2, 13
      ISPRNG(I) = 0
  600 CONTINUE
      CALL GRAPHO(IDEV, ISPRNG, 13,11,IER)
X     IF(IER.NE.0) OUTPUT(101) IER
      CALL GRAPHO(IDEV, ISPRNG, 13,12,IER)
X     IF(IER.NE.0) OUTPUT(101) IER
      RETURN
      END
```

LIST OF REFERENCES

1. Winfrey, R. C., Dynamics of Mechanism With Elastic Links, Research and Development Division, Aerospace Group, Hughes Aircraft Co., Report No. JT 5180, March 1969.

2. Anderson, R. V., Dynamic Analysis of Cams, M. S., ME Thesis U.S. Naval Postgraduate School, 1971.

3. Thomson, W. T., Vibration Theory and Applications, 1st ed., Prentice-Hall, Inc., New Jersey 1965.

4. Newmark, N. M., "A Method of Computation for Structural Dynamics," ASCE, Journal of Engineering Mechanics Division, v. 85, No. EM3, p. 67-94, July 1959.

5. Scientific Data Systems, SDS- FORTRAN IV REFERENCE MANUAL, SDS, Santa Monica, California, 1966.

6. Scientific Data Systems, SDS-9300 Computer Reference Manual, SDS, Santa Monica, California, 1966.

7. Scientific Data Systems, SDS Real Time Monitor Reference Manual, SDS, Santa Monica, California, 1966.

8. Rubinstein, M. F., Matrix Computer Analysis of Structures, 1st ed., Prentice-Hall, Inc., 1965.

9. Hurty, W. C., and Rubinstein, M. F., Dynamics of Structures, 1st ed., Prentice-Hall, Inc., 1966.

10. International Business Machines Corporation Form C28-6515-7, IBM System/360 FORTRAN IV Language, 1968.

11. International Business Machines Corporation Form H20, 0205-3, IBM System/360 Scientific Subroutine Package (360-CM-03X) Version III Programers Manual, 1968.

12. IBM System/360 Operating System, FORTRAN IV (H) Programmer's Guide, Form C-28-6602-1, IBM Corp., Poughkeepsie, N.Y.

13. W. R. Church Computer Center, Users Manual, 1st ed., Rev. 6, March 1970.

14. Mitchell, D. B., "Cam Follower Dynamics", Machine Design, v. 22, p. 151-154, June 1950.

15. Rothbart, H. A., Cams, 1st ed., John Wiley & Sons, Inc., 1956.

16. Stoddart, D. A., "Polydyne Cam Design", Machine Design, v. 25, p. 121-135, January 1953.

17. Molian, S., "Use of Algebraic Polynomials in Design of Cams", The Engineer, v. 215, p. 352-354, 22 February, 1963.

18. Barkan, P., and McGarrity, R. V., "A Spring-Actuated, Cam-Follower System: Design Theory and Experimental Results", Journal of Engineering for Industry, v. 87, p. 279-286, August 1965.

19. Gnilka, C. W., and Bowden, P. K., Dynamic Analysis of a Four Bar Mechanism Utilizing Interactive Computer Graphics, paper presented for M.E. Laboratory Project (ME-3400), Dept. of Mech. Eng., U.S. Naval Postgraduate School, June 1971.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Director, Computer Facility, Code 0211 Naval Postgraduate School Monterey, California 93940	1
4.	Chairman, Deparment of Mechanical Engineering Code 59 Naval Postgraduate School Monterey, California 93940	1
5.	Professor David Salinas, Code 59Zc Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	3
6.	Dr. Richard C. Winfrey Naval Civil Engineering Laboratory Port Hueneme, California	3
7.	Supervisor, Computer Laboratory Code Ec Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
8.	Charles W. Gnilka, LT, USN Philadelphia Naval Shipyard Philadelphia, Pennsylvania 19112	5

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

[ORIGIN]ATING ACTIVITY (Corporate author) [Nav]al Postgraduate School [Mon]terey, California	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

[REPO]RT TITLE

[ ] Modeling and Analysis of an Elastic Mechanism With Clearances

[DESC]RIPTIVE NOTES (Type of report and, inclusive dates)

[Mas]ter's Thesis; December 1971

[AUTH]OR(S) (First name, middle initial, last name)

[Cha]rles W. Gnilka

[REPO]RT DATE [De]cember 1971	7a. TOTAL NO. OF PAGES 342	7b. NO. OF REFS 19
[CO]NTRACT OR GRANT NO. [PR]OJECT NO.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

[DIS]TRIBUTION STATEMENT

[Ap]proved for public release; distribution unlimited.

[SU]PPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940

[AB]STRACT

The classical approach for the analysis of linkages has been to assume [r]igid elements during operation. As the operational speeds of mechanisms [i]ncrease, this assumption creates larger errors in the analysis. Thus a [b]etter model of a machine that considers elastic deformations will aid in [m]ore efficient design as well as provide improved accuracy and performance.

This thesis presents a simulation for the dynamic response of an elastic [l]ink model with three clearances. The simulation has been developed for an [a]utomobile, cam actuated valve train. The computer program is coded in FORTRAN IV, for an IBM-360 computer and is included as a portion of this [w]ork. The capability to visually observe the dynamic action of the model is included by a graphic display routine. This routine is implemented in SDS FORTRAN IV for a SDS-9300 computer interfaced with an ADAGE Graphics Display Terminal, Model 10.

Sample problems for various cam speeds of interest are included utilizing graphs and photos from the graphic display.

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
lastic Mechanism						
am Actuated Valve Train						
omputer Graphics						
learance						
ynamic Modeling						
omputer Simulation						

Thesis 133468
G522 Gnilka
c.1 The modeling and analysis of an elastic mechanism with clearances.